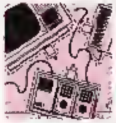
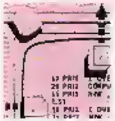
INTELLIVISION® MATTEL ELECTRONICS®

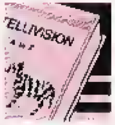
COMPUTER MODULE OWNER'S GUIDE

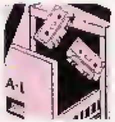
4187-6002

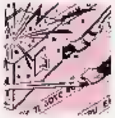
TABLE OF CONTENTS











Introduction

Dear Owner:

You have just made a purchase that will turn your Intellivision® or Intellivision® II Master Component into a Family Entertainment Computer System! "What's a Family Entertainment Computer System?" you ask. Quite simply, it's a whole new world of fun, music, learning and computer power.

You can still play great Intellivision games on your Family Entertainment Computer System — old favorites as well as new Super Games. But now you can do more. You can run children's learning cartridges, music cartridges and special computer software cartridges — all designed to let you interact with your Intellivision® in brand new ways. (For example, you can use BASIC programming commands to change the game play in computer software cartridges (sold separately).

You can play your Intellivision® Music Synthesizer (sold separately) when it is plugged into your Computer Adaptor. In fact, you can play musical notes right on your Computer Keyboard, instead of typing characters...just as a little plus!

You can even create your own computer programs, using the simplified Intellivision BASIC language that's built right into your Computer Adaptor. You can write programs that will help you with your home or business management. Or, on the lighter side, you can create programs for your own video games! Intellivision BASIC offers you special graphic tools that let you extract moving objects (tanks, robots, baseball players, airplanes, that sort of thing) from Intellivision cartridges you already own...then use them to make up your own games.

Intellivision BASIC is not a difficult language to learn. It takes a little attention and a little time on your part, to get familiar with some new concepts. After that, the key is practice. As with any language, BASIC is sometimes easier for children to learn than it is for adults, so don't be afraid to introduce the kids to the Computer Module. Children under 12 may need a little extra assistance from you — the written instructions are geared for bigger people. But overall, the only problem you may have with the kids and the computer is prying the two apart.

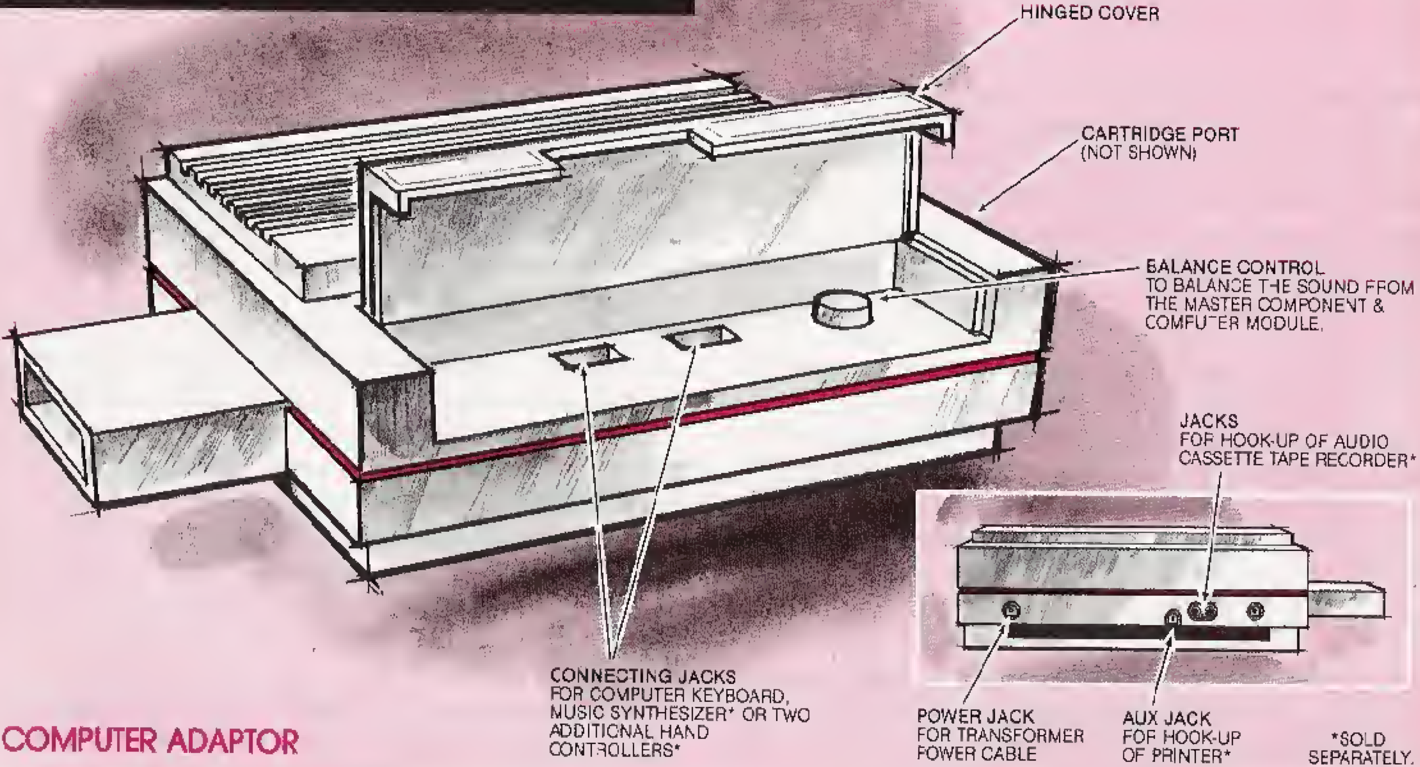
If you are completely new to BASIC programming, you will want to send for your free copy of the book, Step-By-Step Guide To Home Computing. Just indicate on your purchase registration card that you want this book, by checking the appropriate box. Then send in your purchase registration card. Don't forget your name and address, and allow 6 to 8 weeks for delivery.

The Step-By-Step Guide To Home Computing book will introduce you to the most fundamental concepts of computer programming and show you exactly how to use them to build your own computer programs.

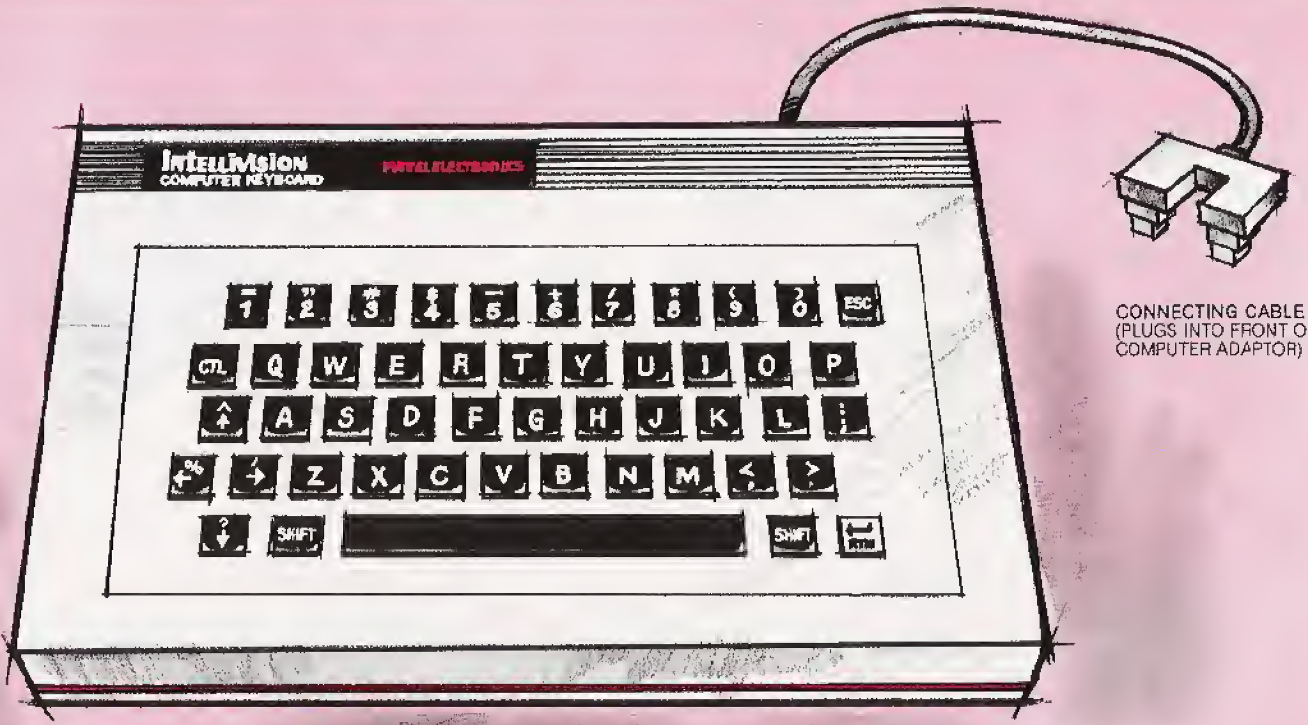
In the Owner's Guide that you are reading now, you will find hook-up instructions, information on the keyboard controls and how they work, a quick preview of computer programming for beginners (including a program you can enter and run), a detailed look at how you can use the special graphics and sound tools, and a slightly technical description of all the commands and "keywords" that are used in Intellivision BASIC. This description is primarily for people who are already familiar with the BASIC language, and want to know specifics about Intellivision BASIC. Once you become familiar with programming, you will also find it a quick and helpful reference.

When all is said and done, the key to using your Intellivision Computer Module lies in remembering that it is just a machine. In the end, like any machine, it can only do what YOU make it do.

YOUR COMPUTER MODULE


HINGED COVER
CARTRIDGE PORT (NOT SHOWN)
BALANCE CONTROL TO BALANCE THE SOUND FROM THE MASTER COMPONENT & COMPUTER MODULE.
JACKS FOR HOOK-UP OF AUDIO CASSETTE TAPE RECORDER*
CONNECTING JACKS FOR COMPUTER KEYBOARD, MUSIC SYNTHESIZER* OR TWO ADDITIONAL HAND CONTROLLERS*
POWER JACK FOR TRANSFORMER POWER CABLE
AUX JACK FOR HOOK-UP OF PRINTER*
*SOLD SEPARATELY.

COMPUTER ADAPTOR

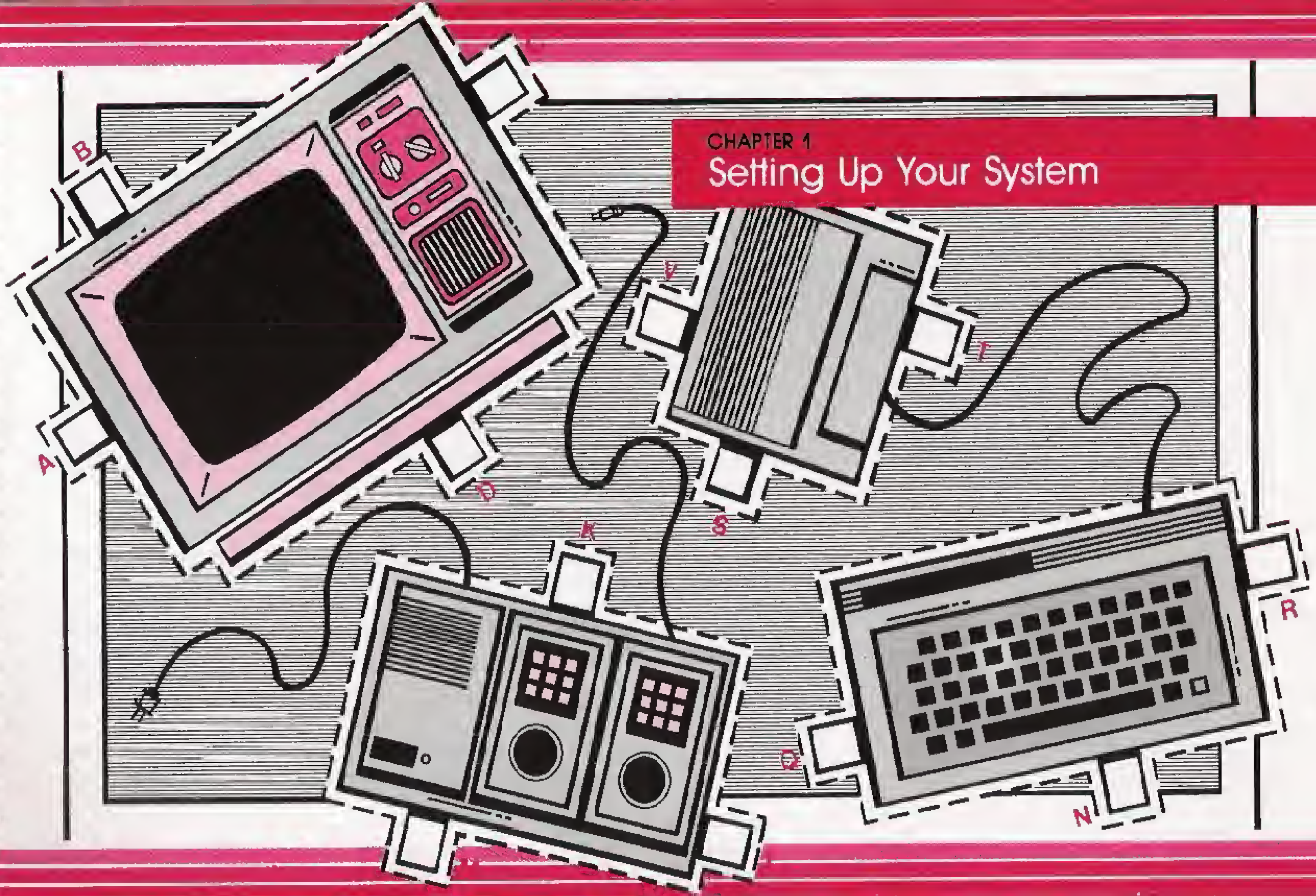


CONNECTING CABLE
(PLUGS INTO FRONT OF
COMPUTER ADAPTOR)

COMPUTER KEYBOARD

CHAPTER 1

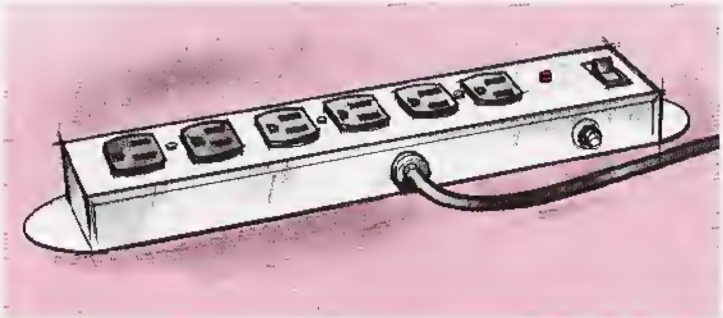
Setting Up Your System



Setting up your Intellivision® Computer Module is as easy as pushing in a few plugs. We'll be doing this step-by-step. You may not yet own all the accessories we will cover in this hook-up instruction. If that's the case, just skip over the steps that don't apply to your system.

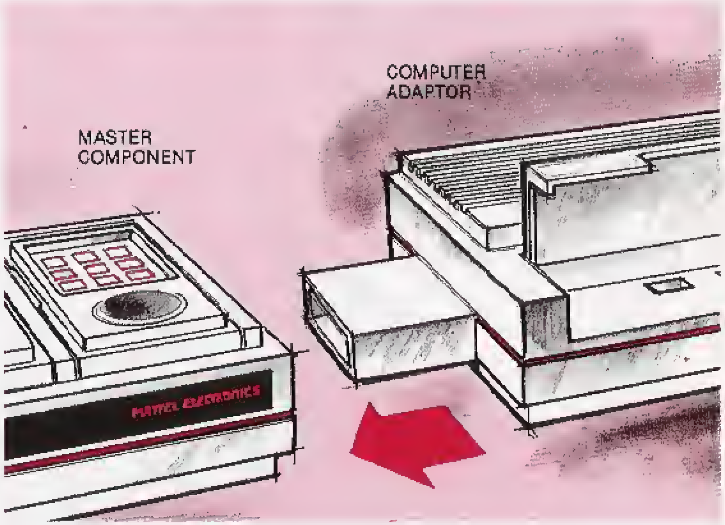
SET-UP WITH INTELLIVISION®

After removing your Computer Module from its package, place it on a flat, sturdy surface, big enough to hold both your Intellivision® or Intellivision® II Master Component and the Computer Module (Keyboard and Adaptor). This surface should also include enough additional space for other accessories (cassette, printer, etc.) that you have or plan on having. If you have additional accessories, you may want to purchase a grounded multiple outlet power strip, to avoid running extension cords to different wall outlets.



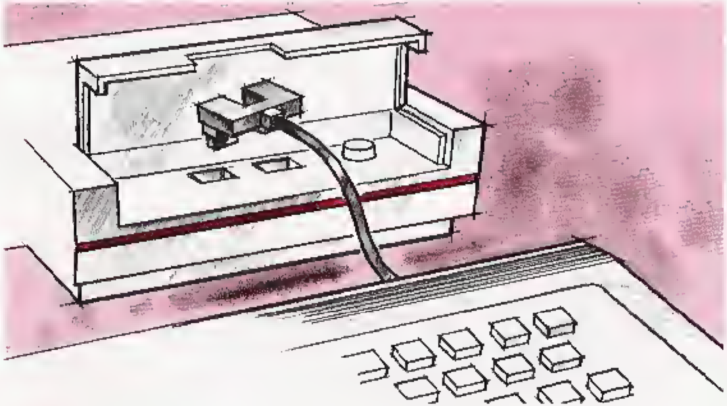
MAKE SURE EVERYTHING IS UNPLUGGED AND SHUT OFF BEFORE YOU START.

Plug your Computer Adaptor into the cartridge port on the right side of the Master Component (where you would normally insert a game cartridge). Push the Adaptor in as far as it will go, so the connection is secure.



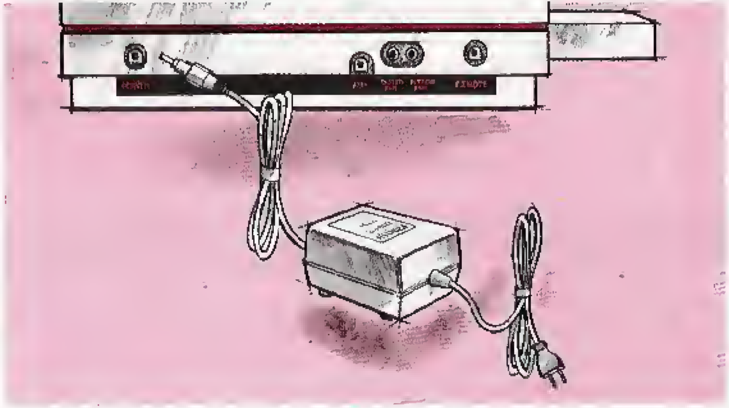
The port on the right side of the Computer Adaptor is where you plug in game cartridges or your Voice Synthesis Module.

On top of the Computer Adaptor, toward the front, is a hinged cover. Lift this cover and you will see two connectors.



These connectors allow you to plug your Computer Keyboard, Music Synthesizer OR two additional cisc or joystick hand controllers into the Computer Adaptor. For now, 'ind the connecting cable on your Computer Keyboard and plug it securely into the connectors on the Computer Adaptor.

Now, let's connect the TRANSFORMER, which supplies power to your Computer Module. Look on the back panel of the Computer Adaptor, where all the other jacks are located. The first jack on the left-hand side is labelled POWER. Insert the pin-plug at one end of the transformer's power cord into this jack. **NOTE:** Use the transformer marked "For use with Intellivision Computer Adaptor only."



The transformer can always stay plugged into a standard 110/120 volt AC wall outlet — but don't plug it in yet. Start everything up in the following order:

1. If you are using an audio cassette recorder or printer with your Computer Module, plug these into a standard wall outlet. (Hook-up to the Computer Adaptor is described on page 11.)

2. Plug the transformer for the Computer Adaptor into a wall outlet.

3. Turn on your television set

4. Plug your Master Component transformer into a wall outlet. Turn the OFF/ON switch to ON and press the RESET button. You will see a title screen with "ECS" and copyright information.

5. Press the DISC on either HAND CONTROLLER. The screen will show a "menu" of three ways you can use your Computer Module. We'll talk about this menu in a little bit.

VOLUME BALANCE

Inside the hinged cover of your Computer Adaptor, toward the right side, is a BALANCE CONTROL knob. This is used to balance the SOUND coming from the Master Component and the Computer Module. To do this:

1. When you see the MENU on your TV screen, choose BASIC by pressing key 1 then ENTER on a HAND CONTROLLER.

2. You will see a blank screen with a square in the upper left corner. Now type in the following program, exactly as it appears below, including spaces. Press the RTN key after each line that you type. If you make a mistake, flip ahead to page 22 for directions on making corrections.

```
10 V=15

20 P=200

30 C=1

40 CALL HUSH

50 CALL TONE

60 PRIN C

70 C=4

80 CALL HUSH

90 CALL TONE

100 PRIN C

110 GOTO 30
```

3. Now type RUN and press the RTN key. When you run this program, you will hear two alternating tones that will probably be different in volume. (If they sound exactly the same, you don't need to adjust the balance.) Move the BALANCE CONTROL knob left or right until the two tones sound alike.

4. Stop the program by pressing the ESC key. This program is now stored in your computer's memory, until you turn your Master Component off or press the RESET button. If you have a cassette recorder hooked up to your computer, you can save the program for future use. You can also use it as a test program, to check the hook-up of your cassette recorder or printer. (See the following sections.)

CASSETTE RECORDER HOOK-UP

This section is for you, if you plan on storing your Computer Module activities on a cassette recorder or using programming software that is stored on cassette. Most good, portable audio cassette recorders with a remote start/stop feature will work just fine with the Intellivision® Computer Module. (The Aquarius™ Data Recorder available from Mattel Electronics® is completely compatible with your Computer Module.)

You will need three special cables for hook-up. The cables are not hard to find. You should be able to pick them up at any good stereo or computer store. This is exactly what you'll need:

1. Two cables with a miniphone plug at both ends.

2. One cable with a sub-miniphone plug at both ends.

The cables should be 3 to 5 feet long for space considerations.

Once you've obtained your cassette recorder and cables, you're ready to go ahead and set it up. First, look at the side of your recorder. See the jacks marked EAR, MIC and REM? Now look at the back of your Computer Adaptor. Find the jacks marked IN FROM TAPE, OUT TO TAPE and REMOTE.

1. Plug one end (it doesn't matter which end) of a mini-plug cable into the jack marked EAR on the cassette recorder. Plug the other end of this cable into the jack marked IN FROM TAPE on the Computer Adaptor.

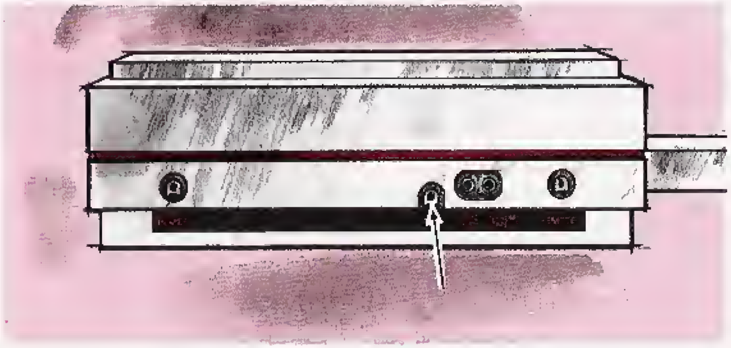
2. Plug one end of the other mini-plug cable into the jack marked MIC on the cassette recorder. Plug the other end of this cable into the jack marked OUT TO TAPE on the Computer Adaptor.

3. Plug one end of the sub-mini-plug cable into the jack marked REM on the cassette recorder. Plug the other end of this cable into the jack marked REMOTE on the Computer Adaptor.

4. Remember the program that you wrote to balance the sound from your Master Component and Computer Module? If this program is still stored in memory, use it to test the operation of your cassette recorder. Follow the directions on page 80 and 81 of Appendix B, to save this program, then verify it and re-load it back into memory. Use "TEST" as your program name for this purpose.

HOOK-UP FOR PRINTER

This section is for those who wish to add a printer to their system. The AQUARIUS™ Printer, available from Mattel Electronics®, is compatible with your Intellivision Computer Module. It comes with a cable that has a single plug at either end. (Consult your Aquarius Printer manual for correct hook-up on the printer side.)



Once you've connected your printer to its cable, look at the back of your Computer Module. There is a jack, right next to the cassette jacks, labelled AUX. Take the remaining pin-plug and insert it into the AUX jack.

Other printers may be compatible with your Computer Module. To find out if your printer is compatible, call one of the Service/Information numbers listed on page 92 of this book.

When you have completed hook-up of your printer, use the program from page 10 that you wrote to balance the sound (if this program is still stored in memory). Simply type:

D = −1 and press the RTN key
CALL OUTP and press the RTN key

A list of the program you wrote should print out on paper.

TURNING YOUR COMPUTER OFF

To keep your Computer Module working properly, follow these steps in turning it off:

1. First turn off your TV set.

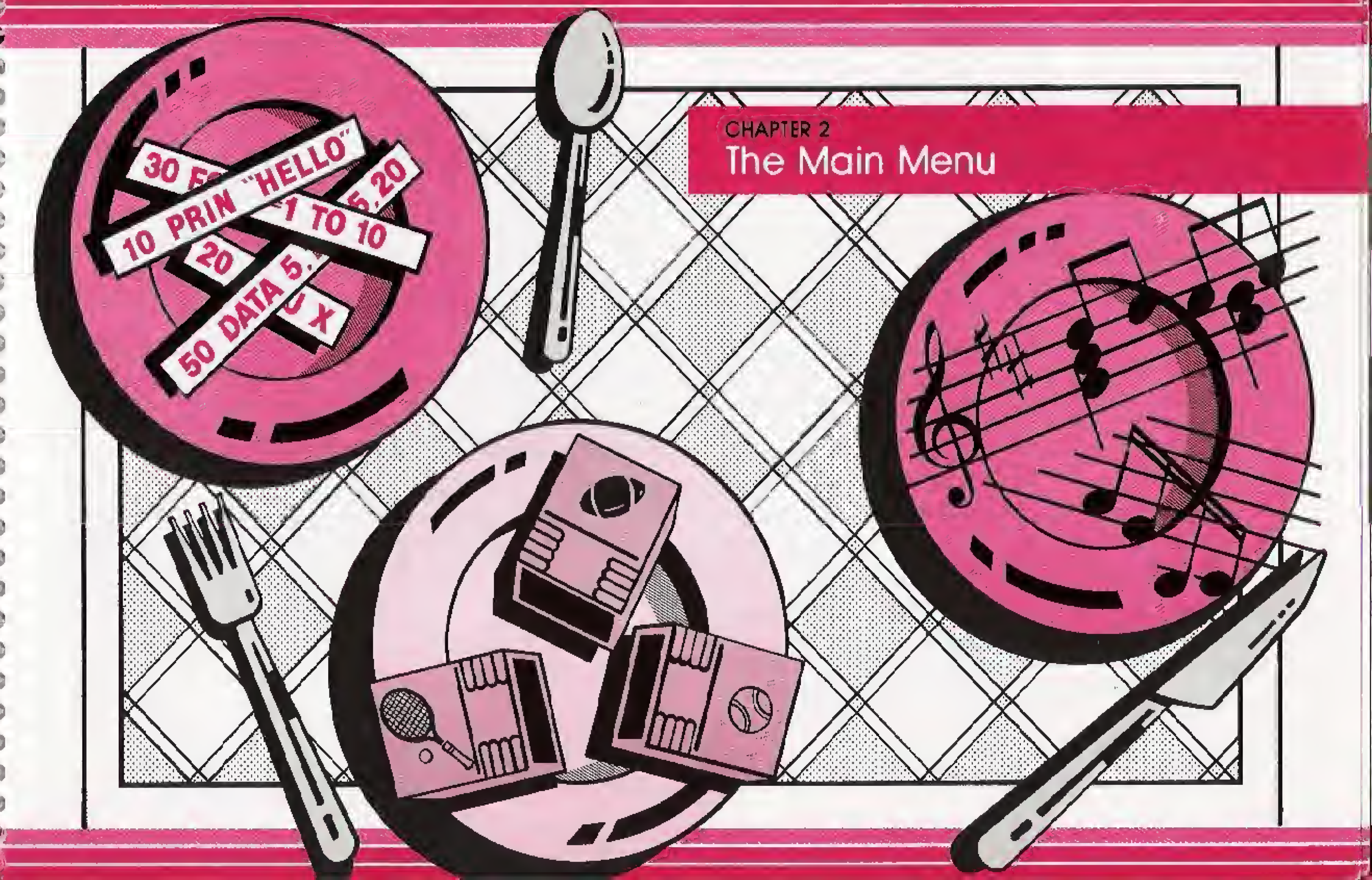
2. Then turn off the power switch on your Master Component.

If you turn your Master Component off first, you may hear a loud hissing noise from your TV set. Don't worry. You haven't broken anything. Go ahead and turn your TV set off.

A WORD TO THE WISE: When you turn your Computer Module OFF, you erase everything you have stored in the computer's memory.

CHAPTER 2

The Main Menu



The list you saw after pressing the Disc on your Hand Controller is called the Main Menu. Anytime you get a list of options or choices, you are getting what is called a menu. Menus allow you to choose from a list of activities or possibilities.

Our menu in this case consists of the three main areas of Computer Module activity:

1. BASIC

Press key 1, then ENTER on either Hand Controller, to write programs using the built-in BASIC language or to extract moving objects from regular Intellivision game cartridges.

2. CARTRIDGE

Press key 2, then ENTER on either Hand Controller, to play an Intellivision or Intellivoice game cartridge or a special Intellivision computer software cartridge.

When you select the CARTRIDGE option, you must have a cartridge inserted in the cartridge port of the Computer Adaptor. When you press key 2 and ENTER, the title of the game will appear on the screen. Follow the directions accompanying your cartridge, to play the game.

3. MUSIC

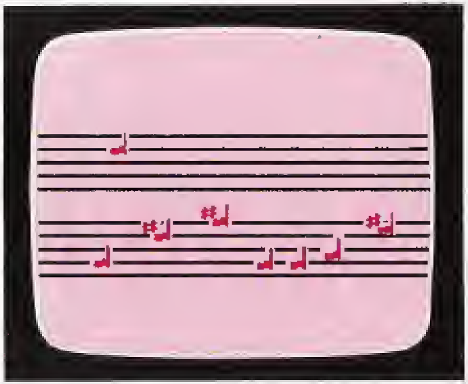
The MUSIC option will enable you to:

■ Use a special music cartridge. These cartridges are designed to be used with the Music Synthesizer, though they can be used with the Computer Keyboard. Insert the music cartridge into the cartridge port of the Computer Adaptor Press key 3 and ENTER on your Hand Controller, then follow the directions that accompany the cartridge.

■ Play the Music Synthesizer, with or without a special music cartridge. When you use the Music Synthesizer, it will be plugged into the same place on the Computer Adaptor where you plugged in the Computer Keyboard.

■ Use the Computer Keyboard to play musical notes. If your Computer Keyboard is plugged into the Adaptor and you select the MUSIC option, it now produces musical notes instead of typed characters. Each different key produces a different musical note.

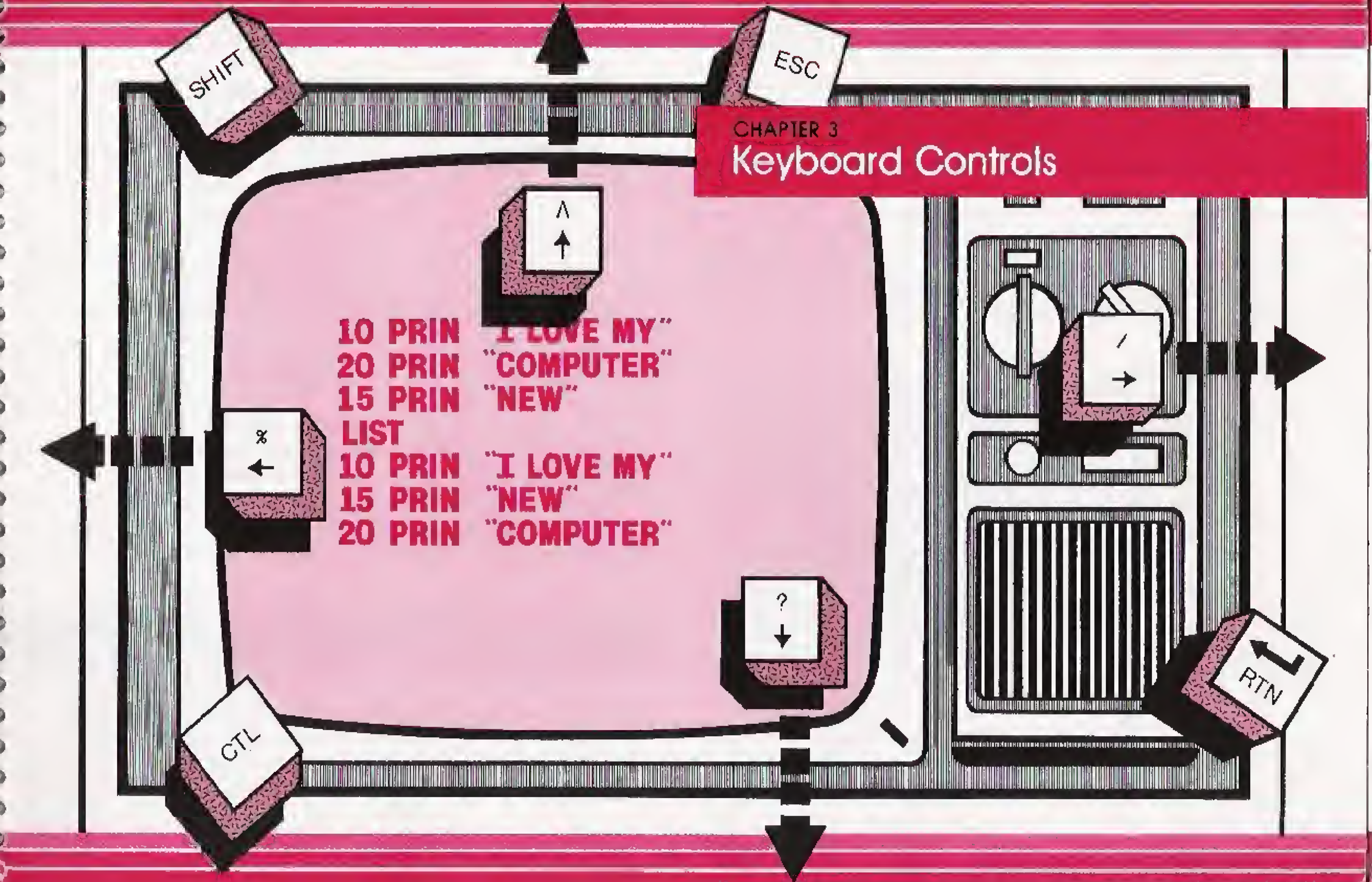
The notes you play appear on the screen, in their proper position on the musical staff lines.



For now, select the BASIC option.

CHAPTER 3

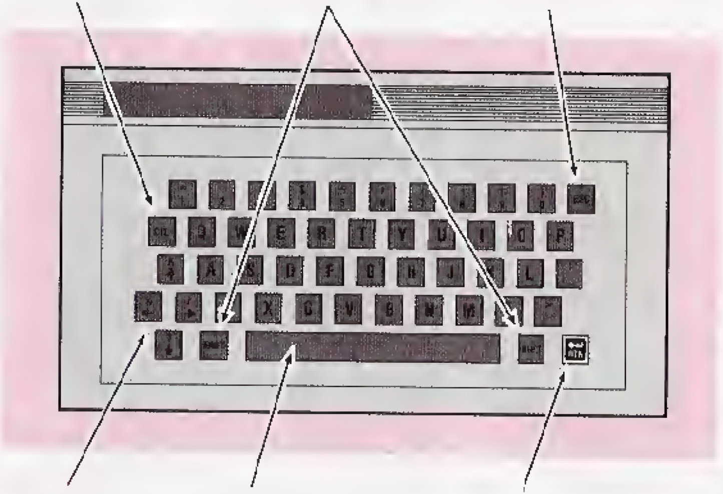
Keyboard Controls



```
10 PRIN  "I LOVE MY"
20 PRIN  "COMPUTER"
15 PRIN  "NEW"
LIST
10 PRIN  "I LOVE MY"
15 PRIN  "NEW"
20 PRIN  "COMPUTER"
```

Everything you do in BASIC is handled through the Computer Keyboard, with results shown on your TV screen. Your Computer Keyboard is your way of communicating with your Computer Adaptor and your Master Component.

Most of the keys on your Computer Keyboard are used in exactly the same way as the keys on a regular typewriter. Some keys, however, have special uses. Take a good look at the keyboard below left, before you read any further.



CTL (CONTROL) — NOT USED WITH BUILT-IN BASIC. RESERVED FOR USE WITH EXTENDED BASIC (AVAILABLE LATER IN 1983).

SHIFT — SAME AS ON A TYPEWRITER, SHIFTS THE KEYBOARD FROM LOWER TO UPPER CASE CHARACTERS.

ESC (ESCAPE) — ALLOWS YOU TO STOP A PROGRAM OPERATION, SUCH AS RUNNING, LISTING OR PRINTING.

ARROW KEYS — MOVE THE CURSOR UP, DOWN, RIGHT OR LEFT.

SPACE BAR — TYPES A BLANK SPACE, WHICH COUNTS AS ONE CHARACTER AND USES MEMORY.

RTN (RETURN) — ACTS AS A CARRIAGE RETURN. ENTERS A COMMAND OR PROGRAM LINE.

Let's take a closer look at two of these keys — RTN and ESC.

RTN (RETURN)

When you have finished typing a command or program line, press the RTN key. This "enters" the command or program line into the computer's memory and also moves the cursor down to the beginning of the next line.

ESC (ESCAPE)

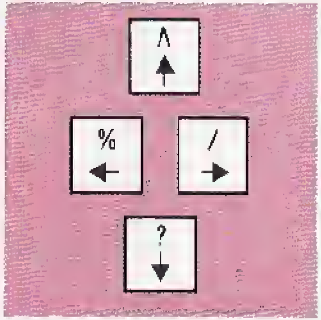
Press ESC when you want to stop a program that is running. This is particularly useful if you run a program that contains a perpetual loop (one that repeats endlessly). You can also use ESC to stop other operations on a program, such as listing a program, printing out a program, storing a program on cassette, etc.

CURSOR AND CHARACTERS

See the little square at the upper left corner of the screen? This is called the CURSOR. Anytime you press a letter, number or symbol key, this square moves one space in front of the character you typed. Thus it marks the place where the next character you type will appear. The cursor takes up the same space as one character. A character is any letter, number or symbol, or a single space between two other characters.

MOVING THE CURSOR

Now look at the ARROW KEYS on your keyboard.



These keys move the cursor in the directions they point: up, down, right or left. The space bar also moves the cursor to the right, but in addition types a space, which counts as a character and uses memory. The arrow keys are specifically for moving the cursor on the screen without typing a space or using memory.

CHARACTERS

Always keep in mind that NOTHING is interchangeable on a computer keyboard, the way it is on a regular typewriter keyboard. You cannot, for instance, substitute a letter l for a number 1. A group of characters enclosed in quotation marks is commonly called a STRING. Strings are used in both commands and program statements.

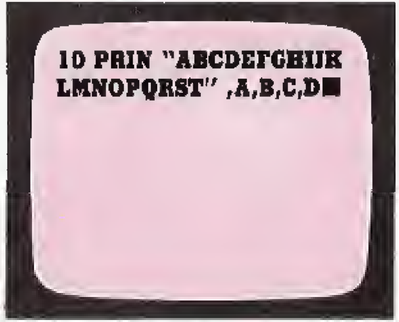
LINE LENGTH

A single keystroke for a letter, number, symbol or space equals one character. Thirty-nine characters equal one line. Lines are structured like this:

1. You can type twenty characters across the screen

2. If you continue typing, without pressing the RTN key, nineteen more characters will print out beneath the first twenty. This is called "wrapping around".

3. These maximum thirty-nine characters are considered a single line by the computer. For example, if you were attempting to write a line in a BASIC program, you could write a TOTAL of thirty-nine characters before pressing the RTN key and ending the line. (Remember that spaces also count as characters.)



```
10 PRIN "ABCDEFGHIJK
LMNOPQRST" ,A,B,C,D■
```

Once you have typed thirty-nine characters, the computer will refuse to print any more characters until you press the [RTN] key.

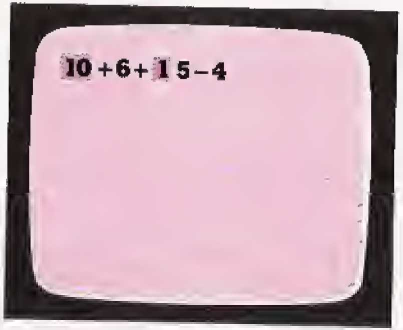
MAKING CHANGES OR CORRECTIONS

If you want to go back and change something you have already typed, use the ARROW KEYS to move the cursor into position over the character you want to change. Then type in the new character or characters you want. Be sure you re-type everything to the right of your change, or use the RIGHT ARROW KEY to move the cursor to the end of the line, BEFORE you press [RTN]. If you leave the cursor in the middle of a line, after making a change, everything to the right of the cursor will be erased when you press [RTN]. (This is called TRUNCATING.)

Example:

- Type "10 + 6 + 15 − 4" and press [RTN].

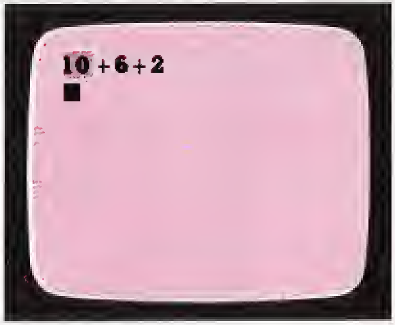
- Use the UP and RIGHT ARROW KEYS to position the cursor over the 1 in 15.



```
10+6+15-4
```

- Type a 2 in place of the 1. Now press [RTN].

- You will now see "10 + 6 + 2". The characters "5 − 4" have been truncated.



```
10+6+2
■
```

The arrow keys can also be used for other, more sophisticated editing jobs. See Appendix C, page 86 for details.

CLEARING THE SCREEN

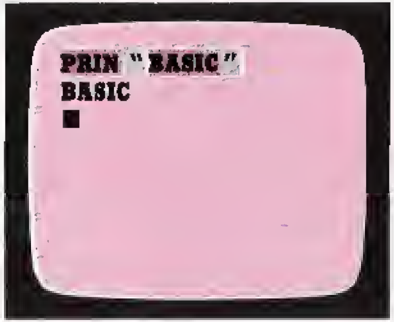
To erase everything on the screen and return the cursor to the upper left corner (called the "home" position), type CLR and press RTN. CLR does not erase the memory, only the screen.

CLR can also be used as a statement in a program, causing the screen to be cleared each time the computer reads this instruction.

"BASIC" MISTAKES — HOW TO RECOGNIZE THEM

If you type a BASIC command or program statement incorrectly, your computer will let you know immediately.

Each different part, or element, of a BASIC command or statement is color-coded. (Program statements are explained in the next chapter.) When you press RTN at the end of the command or statement, each element that the computer recognizes and is able to act on turns its specific color.

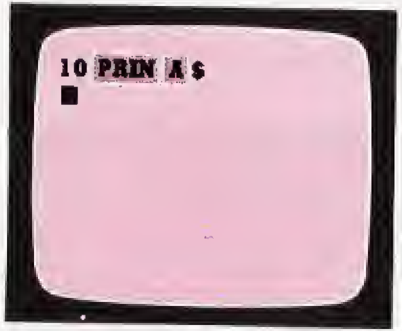


```
PRIN "BASIC"
BASIC
■
```

Certain commands, called MONITOR COMMANDS (RUN, LIST, NEW, DEL and a few others) are exceptions to this rule. Monitor commands do not change color when you press RTN. (Monitor commands are explained on page 29.)

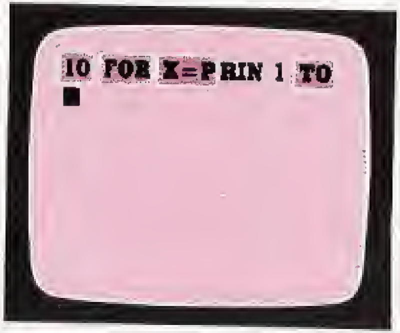
If you have typed an element incorrectly or in the wrong order, the computer may not recognize the element, or recognize it but not be able to act on it.

If the computer does not recognize a single element but can still carry out the command or statement, it will color everything in the line that it can carry out and leave the rest uncolored.



```
10 PRIN A $
■
```

If the computer does not recognize enough of a command or statement to carry it out, then it will turn as much of the line as it recognized gray. The rest will not turn color at all.



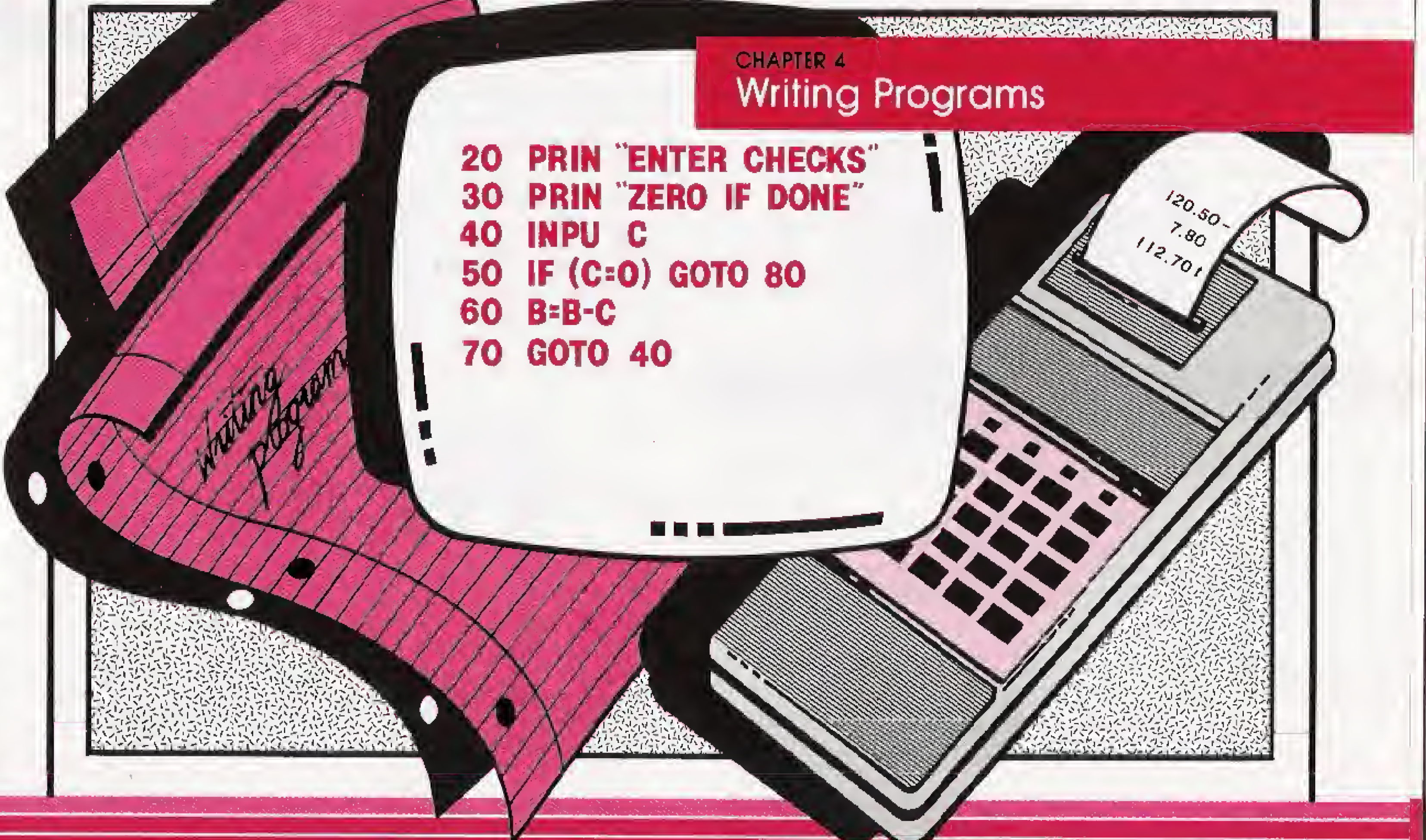
```
10 FOR X=PRIN 1 TO
■
```

So as soon as you enter a command or statement, you know if there's a problem, and exactly where the problem lies.

There is a list of the different ways in which you can make a mistake and the color codes for different elements in Appendix C, pages 84-86.

CHAPTER 4

Writing Programs



```
20  PRIN "ENTER CHECKS"
30  PRIN "ZERO IF DONE"
40  INPU  C
50  IF (C:0) GOTO 80
60  B=B-C
70  GOTO  40
```

If you are new to programming, this chapter will give you a taste of Intellivision BASIC. BASIC is a computer language developed for people like you, whose programming needs are not particularly scientific or heavily financial. The word BASIC stands for Beginner's All-Purpose Symbolic Instruction Code. That means it's easy to use for many different purposes.

This chapter will introduce you to a few of the things you can do with Intellivision BASIC, as well as some "basic" concepts. It is not a course in BASIC programming. For that, you need to send for the book, Step-By-Step Guide To Home Computing. (See page 3.)

THE STRUCTURE OF A PROGRAM

A program is a list of very specific instructions that give the computer a certain job to do.

Each separate instruction in the list is called a STATEMENT. A statement can be any length, up to the maximum line length (39 characters). Each statement is therefore a separate LINE in the program and is identified by a LINE NUMBER.

EXAMPLE: 10 PRIN "HELLO" is a statement.

When you finish typing a statement, you "enter" it into the computer's memory by pressing the RTN key.

LINE NUMBERS

A line number can be any whole, positive number from 0 to 31999. It appears at the beginning of the line. Line numbers do two very important jobs:

1. They tell the computer the ORDER in which the statements in a program should be carried out, or "executed". The computer always executes program statements in order of the lowest to the highest line numbers...regardless of the order in which you enter them. For example, if you enter a series of statements numbered in this order:

```
40
10
30
20
```

They will be re-ordered and executed in this order:

```
10
20
30
40
```

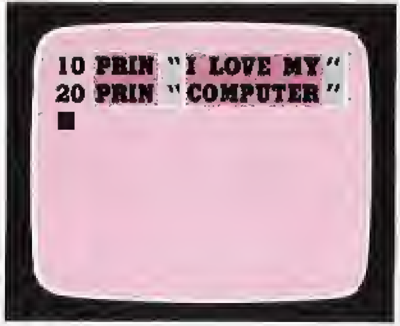
2. They alert the computer to the fact that an instruction is part of a program...that is, a statement. Numbering a statement is the same as saying, "Wait. See what happens next. We're building something here." If you don't number a statement, the computer tries to execute it as soon as you press RTN.

When you number program statements, it is a good idea to work with increments of 10. That way you have room between program statements to add in other statements.

ADDING A PROGRAM STATEMENT

To add a statement to a program, select a line number that falls between the numbers of the statements immediately before and after the point where you wish to insert the statement.

Example: You wish to insert a PRINT statement between lines 10 and 20 of this short program.

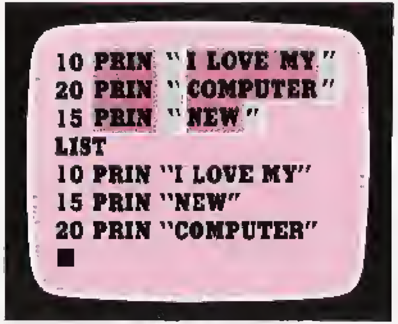


```
10 PRIN "I LOVE MY"
20 PRIN "COMPUTER"
■
```

Type the following line and press RTN:

15 PRIN "NEW"

Type LIST and press RTN, to display your new program.



```
10 PRIN "I LOVE MY"
20 PRIN "COMPUTER"
15 PRIN "NEW"
LIST
10 PRIN "I LOVE MY"
15 PRIN "NEW"
20 PRIN "COMPUTER"
■
```

Notice that Line 15 has been inserted between lines 10 and 20.

The number you use for your new statement must not be used anywhere else in the program, or the new statement will replace the old statement.

DELETING PROGRAM STATEMENTS

Use the DEL (Delete) command to delete statements. To delete a single statement, type DEL followed by the line number for that statement. Then press RTN. Example: DEL 10 (and press RTN)

To delete a group of statements, type DEL, the beginning line number, a comma, then the ending line number. Then press RTN. Example: DEL 10,20 (and press RTN)

To delete everything BEFORE a particular statement, say line 50, type: DEL 0,50 (even though you don't have a line 0).

To delete everything AFTER a particular statement, say line 50, type: DEL 50,1000 (or any very large line number, up to 31999)

CHANGING PROGRAM STATEMENTS

If you wish to make major changes in a program statement, use the arrow keys as described on page 22. Or simply re-type the entire statement, keeping the same line number.

BASIC KEYWORDS

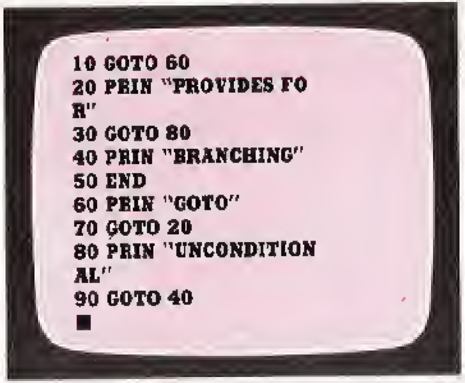
There are several different elements that can combine to make up a program statement. One of the most important of these elements is called a KEYWORD. This is simply a BASIC word. It is used to tell the computer what to do with the other elements in a statement: the numbers, "strings" (remember strings from page 21), variables (values in a program that can change), routines and functions. You will meet these other elements later, some in this book and others in the Step-By-Step Guide To Home Computing.

For a brief description of all BASIC keywords, refer to Appendix A, starting on page 48. For detailed information on how to use each keyword, send for your copy of the book, Step-By-Step Guide To Home Computing.

RUN AND LIST

Executing a program is also known as RUNNING the program. When you want the computer to execute a program, you give it a RUN command. You type RUN and press RTN.

You can also look at your program without running it. This is called LISTING the program. When you list a program, it is displayed on the screen with all statements in the order of their line numbers.



```
10 GOTO 60
20 PRIN "PROVIDES FO
R"
30 GOTO 80
40 PRIN "BRANCHING"
50 END
60 PRIN "GOTO"
70 GOTO 20
80 PRIN "UNCONDITION
AL"
90 GOTO 40
■
```

You can list an entire program or only part of a program. You can even list a single statement in a program.

To list an entire program, type LIST and press RTN. To list a single statement, type LIST, then the line number for that statement, and press RTN. (Example: LIST 10) To list a group of statements within a program, type LIST, the beginning line number for the group, a comma, the ending line number, and press RTN. (Example: LIST 10,50) To stop a program listing once you have started it, press the ESC key.

Notice that when you give the computer a RUN or LIST command, you do not put a line number in front of the command. Notice also that we have been referring to these two instructions as COMMANDS rather than statements.

The reason is simple: A command is not part of a program. It is an instruction that is meant to be executed as soon as you press RTN, not stored in memory for later use.

You can type a program statement without a line number, if you want to preview it, outside of a program. Leaving the line number off causes a statement to be executed as soon as you press RTN.

MONITOR COMMANDS

The commands used most often are the ones that tell the computer to do something with a program that is already written. These are also called MONITOR COMMANDS. In addition to RUN and LIST, monitor commands include NEW, which tells the computer to erase everything currently stored in memory; DEL, which tells the computer to delete statements in a program (see page 28); CSAV, CLOD and CVRF, which are used in saving and loading programs with a cassette recorder (see Appendix B); and MENU, which lists different programming options.

TRYING IT OUT: A BASIC PROGRAM

We're going to show you an easy BASIC program and how it runs. It's a simple checkbook program, used for figuring and maintaining a checkbook balance. Type it in, line by line, exactly as it appears here, including spaces. Then type RUN and press RTN to try it out.

When you run this program, the computer will print out a message asking you to ENTER BALANCE (the ending balance on your monthly statement). It will wait for you to type in the amount of the balance and press RTN, then print a second message, "ENTER CHECKS. ZERO IF DONE". ("Checks" means outstanding checks or other withdrawals.) The computer will wait again while you type in the amounts of your outstanding checks. After each amount that you type, press RTN. When you have no further checks to enter, type a zero and press RTN. (That's what "ZERO IF DONE" is telling you to do.)

The computer will then print out a message asking you to ENTER DEPOSITS and wait for you to type in the amounts of deposits not noted on your statement. After each amount that you type, press RTN. When you have no further deposits to enter, type a zero and press RTN. The computer will then print out your balance and the program will automatically stop.

Before you enter the program below, type NEW and press RTN, to clear the memory. MAKE SURE YOU PRESS RTN AFTER TYPING EACH STATEMENT.

PROGRAM

```
10 INPU "ENTER BALANCE",B
```

Waits until you enter amount of current balance, then stores it as variable B.

```
20 PRIN "ENTER CHECKS"
```

Prints message "ENTER CHECKS"

```
30 PRIN "ZERO IF DONE"
```

Prints message "ZERO IF DONE"

```
40 INPU C
```

Waits for you to enter amount of outstanding check, then stores it as variable C.

```
50 IF (C=0) GOTO 80
```

Looks to see if you have entered a zero. If you have, sends the computer ahead to line 80.

```
60 B=B-C
```

Subtracts the last check entered from the balance and stores the new balance.

```
70 GOTO 40
```

Sends the computer back to line 40, so you can enter another check if desired.

```
80 PRIN "ENTER DEPOSITS"
```

Prints message "ENTER DEPOSITS".

```
90 PRIN "ZERO IF DONE"
```

Prints message "ZERO IF DONE"

```
100 INPU D
```

Waits for you to enter amount of deposit and stores it as variable D.

```
110 IF (D=0) GOTO 140
```

Looks to see if you have entered a zero. If you have, sends the computer ahead to line 140.

```
120 B=B+D
```

Adds the last deposit entered to balance and stores the new balance as variable B.

```
130 GOTO 100
```

Sends the computer back to line 100 to see if you have any more deposits to enter.

```
140 PRIN "THE BALANCE IS",B
```

Prints "THE BALANCE IS" followed by the current value of variable B, which is the current balance.

If your program doesn't run the way it should, LIST it and check each line to make sure you haven't made any mistakes in typing. If you find a mistake, use the arrow keys to go back and correct it. (See page 22.)

When you are through running your program, you have three choices: You can run it again; you can save it on cassette tape (see Appendix B for instructions); or you can erase it from memory with the NEW command.

MENU COMMANDS

Once you've become more familiar with the Intellivision® BASIC vocabulary and you need a quick reference, there's a series of BASIC commands that gives you a listing of all the words in the BASIC vocabulary, right on your TV screen. That series is the MENU series.

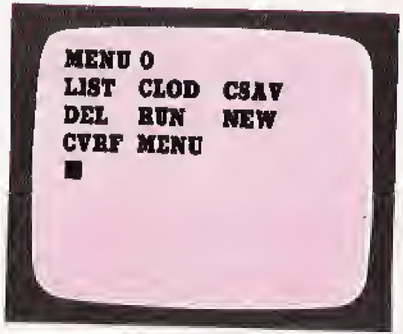
Remember when we talked about the Main Menu? What you get when you enter a MENU command is essentially another menu, sometimes called a submenu because it comes under the heading of an item on the Main Menu (in this case, BASIC).

Below are the MENU series commands. Type these without a line number.

MENU SERIES COMMANDS

Type MENU 0 or just MENU and press [RTN].

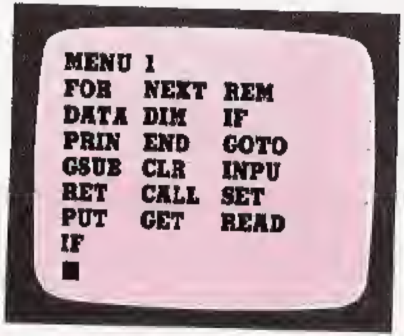
You get a list of MONITOR COMMANDS. These are primarily used to tell the computer to do something with a program, such as run it, list it, etc. Monitor command include RUN, LIST, NEW, CSAV, CLOD, CVRF, DEL and MENU.



```
MENU 0
LIST  CLOD  CSAV
DEL   RUN   NEW
CVRF  MENU
■
```

Type MENU 1 and press [RTN].

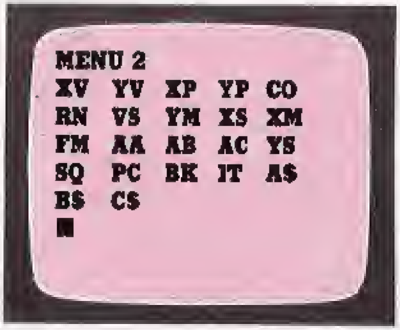
You get a list of BASIC KEYWORDS, such as PRIN, GOTO, INPU, etc. These are the "building blocks" with which you construct program statements.



```
MENU 1
FOR   NEXT  REM
DATA  DIM   IF
PRIN  END   GOTO
GSUB  CLR   INPU
RET   CALL  SET
PUT   GET   READ
IF
■
```

Type MENU 2 and press [RTN].

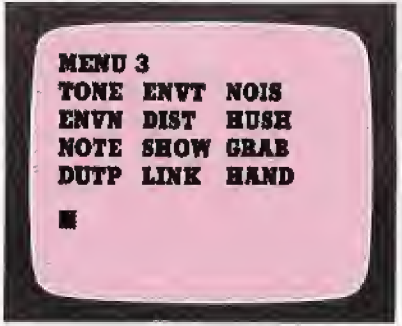
You get a list of BASIC functions.



```
MENU 2
XV   YV   XP   YP   CO
RN   VS   YM   XS   XM
FM   AA   AB   AC   YS
SQ   PC   BK   IT   A$
B$   C$
■
```

Type MENU 3 and press [RTN].

You get a list of BASIC routines.



```
MENU 3
TONE  ENVT  NOIS
ENVN  DIST  HUSH
NOTE  SHOW  GRAB
DUTP  LINK  HAND

■
```

CHAPTER 5

Color, Motion & Sound

Intellivision® BASIC is uniquely designed to allow you to do special things with color, moving objects (for example, the players in a Baseball cartridge), and sound. It has many built-in functions and routines that allow you to access and "make a copy of" moving objects from your regular Intellivision® game cartridges, then change the shape, color and movement of those objects. The "new" objects can be stored in memory and used in programs that you write, to create your own games!

Before we see how this works, let's look at two terms with which you may be unfamiliar — functions and routines.

FUNCTIONS

A function is a way of representing a number value that has a special significance. The functions we will be looking at in this section all have a special significance in regards to a moving object.

A function always contains a LABEL that identifies it and a NUMERIC VALUE IN PARENTHESES that identifies the object to which it relates. Example: CO(5) is a function. CO is the label for the COLOR function. (5) identifies object 5. By setting CO(5) equal to a number value for a certain color, we can change the color of object 5.

Example: CO(5) = 2 turns object 5 red. (2 is the number code for red.)

There are many other functions besides color — many other ways in which we can change the characteristics of an object We will look at a few of them in this chapter. For a complete list of all functions, see Appendix A, starting on page 58.

ROUTINES

A routine is a kind of "mini-program" that has already been written into the computer in assembly language (a mathematical language that is more powerful than BASIC, but difficult for people to use). A routine lets you do something special that you would not be able to do using the BASIC language alone. You don't need to know assembly language to use a routine. You only need to know the name of the routine, how to access it, and how to set certain values that it uses.

To access a routine, you use the keyword CALL followed by the four-letter name of the routine. For example: CALL TONE accesses the TONE routine, which makes the computer generate a sound.

Before the computer can generate a sound, however, it needs to know what kind of sound to generate. It needs to know volume, period (this sets the pitch of the sound — whether it is a high or low sound), and the channel through which the sound is generated. You give the computer this information by setting number values for each piece of information needed. You do this by writing equations that

USING THE SHOW ROUTINE

First, insert a game cartridge into the cartridge port. Make sure that you have selected BASIC option (NOT CARTRIDGE) from the Main Menu.

You can now use the SHOW routine to display a moving object from the cartridge you inserted, through a sort of "blind search" method. Here's how it works.

First you define your moving object. This is a 3-step process.

1. Assign the object a number from 0 to 7. Do this by typing the equation, O = (a number from 0 to 7). (O is the letter O, which stands for Object.) Then press RTN.

Example: O = 1

2. State whether the resolution is single or double, by typing one of these two equations, then pressing RTN:

D = 0 or 1   for single resolution
D = 2   for double resolution

3. Select a picture from the 128 possible pictures, by typing the equation, N = (a number from 0 to 127). Then press RTN.

Example: N = 6

If your resolution is double, you will actually get two pictures instead of one. N = 6 will give you pictures 6 and 7, picture 6 on top and picture 7 on the bottom.

NOTE: IF YOU DO NOT SELECT A NUMBER VALUE FOR OBJECT, RESOLUTION OR PICTURE, THE COMPUTER WILL AUTOMATICALLY SET THAT VALUE TO ZERO (OR ANY NUMBER VALUE THAT HAS PREVIOUSLY BEEN SET AND NOT ERASED).

Second, you tell the computer to display the moving object that you just defined. Type the command CALL SHOW and press RTN. The object you defined will appear on the screen.

STRANGE OBJECTS

If the object you have displayed looks particularly strange, try changing the numbers you used for picture and resolution. You see, any given game cartridge will have pre-set the resolution and picture numbers for all moving objects from 0 to 7. You can reset the picture by typing different numbers into the picture equation. However, you cannot reset the resolution, since ONLY a certain amount of memory space is set aside for each moving object.

If you select double resolution for an object and only single resolution space was reserved for that object, then only half of the object will be displayed. If you have more than one object on screen at the time, the second half of your object may replace the top half of another object.

look like this:

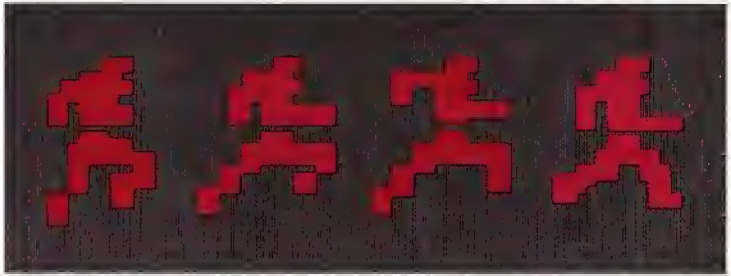
P = 200 (P stands for period)
V = 15 (V stands for volume)
C = 1 (C stands for channel)
CALL TONE

You select the values for each piece of information from a range of possible values. The range for each value in a routine is given in Appendix A, starting on page 68. In this chapter, we will look closely at a couple of routines. For a complete list of available routines, see Appendix A.

FUN WITH MOVING OBJECTS

In any Intellivision game cartridge, there is a maximum of eight moving objects that you can display. These are numbered 0 to 7. You can display these objects on the screen, using one of two routines — SHOW or GRAB.

To understand how to use these routines, you need to first understand how moving objects are created. In its memory your Intellivision® Master Component has stored 128 different pictures. These pictures are used, either alone or in combination, to define all moving objects. A game cartridge contains specific instructions that tell your Intellivision which pictures to use for each moving object in a game. Since an object may change shape when it moves, different pictures may be needed to show different stages of movement (the same way in which animation works).



The game cartridge also tells your Intellivision to set aside a certain amount of memory space for displaying each object. The amount of space depends on the RESOLUTION of the object, which is a way of measuring the object. An object can be single or double resolution. A single resolution object has EITHER a top OR a bottom half. A double resolution object has BOTH a top AND a bottom half. A double resolution object will not fit into the space reserved for a single resolution object (though a single resolution object will fit into the space reserved for a double resolution object).

function in Appendix A (starting on page 58).

You can display more than one object on the screen at one time, but you must use the XP or the YP function to move the first object out of the way before you display the second object (or the second will appear on top of the first object.)

When you use the SHOW routine to display a moving object, you can change the object's position on the screen, but you cannot make it appear to be animated. This is because SHOW only allows you to assign one picture for an object (two for a double resolution object). To animate an object, you must assign a SEQUENCE of pictures to the object. To do this you must use the GRAB routine.

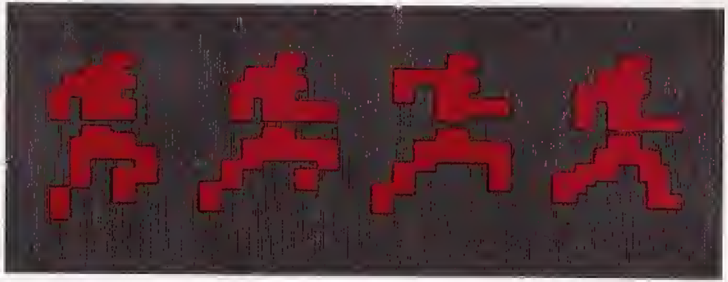
USING THE GRAB ROUTINE

GRAB is very similar to SHOW, with two important differences:

1. GRAB uses memory space and SHOW does not.

2. GRAB allows you to assign a sequence of pictures to an object, so you can animate the object. SHOW does not.

To use the GRAB routine, you must first define the moving object. Follow the same steps that you used in defining an object for the SHOW routine...but add one additional piece of information.



Before you can animate an object, you must assign it a sequence of pictures. Each picture in the sequence will show the object in one of the stages of animation.

The number you select for N (the picture) tells the computer where the sequence starts. You must also give a number to tell the computer how many pictures there are in the sequence. There can be a maximum of 16 pictures in a motion sequence, numbered from 0 to 15. (So if you set M equal to 5, you are assigning 6 pictures to the sequence.) To set the number of pictures in a sequence, type the equation:

M = (a number from 0 to 15)

Example: O = 1

D = 4

```
N = 0

M = 3

CALL GRAB
```

This example tells the computer to assign 4 pictures, starting at picture 0, to moving object 1, displayed in single resolution. When the object is set in motion, it will cycle through 4 pictures.

FUNCTIONS AND MOTION

All of the functions that work with an object displayed by SHOW also work with an object displayed by GRAB. Use the GRAB routine to display a moving object, then try out these additional functions. Remember to replace the 0 within parentheses with the number of your moving object.

TYPE THIS:	TO DO THIS:
PC(0) = a number between 0 and 15	Display a particular picture in the motion picture sequence for an object.
SP(0) = 20	Set the speed with which object moves through its motion sequence.
XV(0) = 40	Set the speed at which object moves horizontally across screen and set it in motion To stop the motion, set the XV function equal to zero.
YV(0) = 10	Set the speed at which object moves vertically across screen and set it in motion. To stop the motion, set the YV function equal to zero.

You can change the number on the right side of the function equations above, within a certain range. The range for each function is given in Appendix A. starting on page 58.

PLAYING WITH SOUND

There are seven routines that deal with sound. You've already met the TONE routine (on page 36). This routine causes a single tone to be generated by the computer When you want the tone to stop, use another, even simpler routine — HUSH. The HUSH routine doesn't require that you enter any additional information. Just type CALL HUSH and press [RTN], and the sound will stop.

Another routine, NOTE, is used with the TONE routine, to set the period of a tone to that of a pre-set note on a musical scale. This determines the exact pitch of the tone generated. It does not itself generate a tone. You can choose from 95 different pre-set notes, to which you can set the period of a tone. Use the equation:

N = (a number from 1 to 95)

First type the equation and press [RTN]. Then type CALL NOTE and press [RTN]. Below is a short program that combines the TONE and NOTE routines to generate a 13-note musical scale. Type in the program exactly as it appears below. Press [RTN] after each line that you type. When you are finished typing the program, type RUN and press [RTN].

```
10 CALL HUSH

20 C=0

30 V=15

40 FOR N=12 TO 24

50 CALL NOTE

60 CALL TONE

70 NEXT N

80 CALL HUSH
```

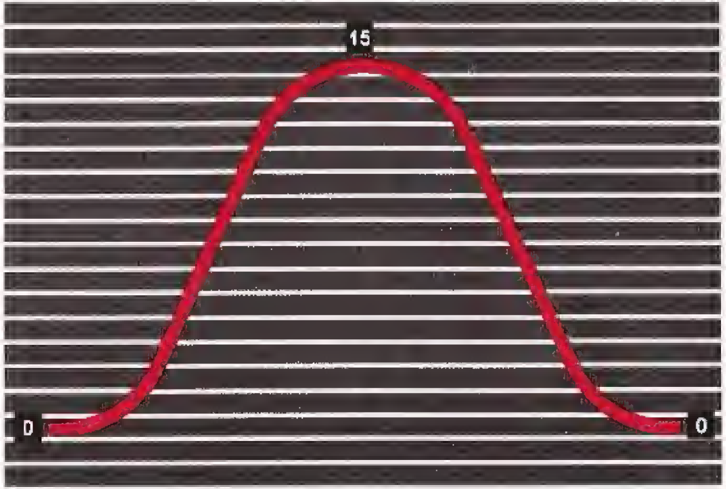
Now type a new line 40 that reads: 40 FOR N=24 TO 36 and press [RTN]. RUN the changed program and listen. You will get a different musical scale this time, one octave higher than the original scale.

ENVELOPE ROUTINES

When you use the TONE routine, you set a definite, unvarying volume for the tone you generate. Unless you change the volume, it remains the same. When you do change the volume, it changes abruptly.

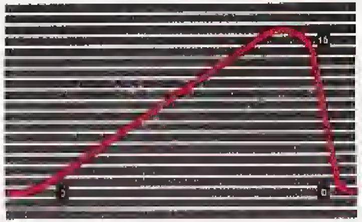
There are two routines called ENVELOPE routines that let you manipulate volume more delicately. These routines break volume down into two parts, called ATTACK and DECAY.

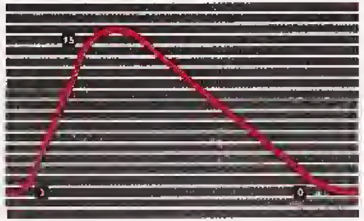
Picture volume as a curve. At the top of the curve is the peak volume. At either end of the curve is volume 0.



If this tone were generated, it would start at 0, rise to 15, then drop back to 0 again. The rise from 0 to 15 is the attack. The drop from 15 to 0 is the decay. The way in which the tone attacks and decays is called the ENVELOPE of the tone.

There are many things that you can do with the envelope of a tone. You can make the attack gradual and the decay sharp, like the graph shown below at left.





You can make the attack sharp and the decay gradual, like the graph shown above at right.

You can make both attack and the decay either gradual or sharp. Or you can set a sharp or gradual attack and then sustain the volume at peak. You do all these things by setting the envelope in an envelope routine. You can also set the overall length of the sound as it attacks and decays, as well as the period of the sound (which determines its pitch) and the channel through which it will be generated.

You can set an envelope for either a tone or for a non-musical sound called "noise". You know what a tone is. "Noise" is the type of sound that the ocean surf or highway traffic or a crowded auditorium produces. ("Noise" is good for producing percussion sounds.) There is a separate envelope routine for tone and for noise.

ENVT (Envelope Tone)

Use this routine to generate a tone whose envelope you want to control. As with the TONE routine, you must define the tone before the computer can generate it. Here's how.

1. Set the channel. There are 6 channels, numbered 0 to 5. Use the equation: C = (a number from 0 to 5)

2. Set the period. This can be a number from 0 to 4095. The higher the number, the lower the pitch of the note.

Use the equation: P = (a number from 0 to 4095)

3. Set the length of the envelope (how long it will take the tone to rise or fall in volume). This can be a number from 0 to 65,000. (Try a number around 5000 to really hear how the envelope can be changed.)

Use the equction: L = (a number from 0 to 65,000)

4. Set the attack/decay characteristics of the envelope, whether sharp or gradual. This can be one of certain numbers between 0 and 15. Each number produces specific characteristics. Look on page 69 of Appendix A for a list of attack/decay numbers.

Use the equation: E = (the characteristic number)

5. Type CALL ENVT and press RTN.

ENVN (Envelope Noise)

Use this routine in exactly the same way as you use the ENVT routine, with these differences:

1. The PERIOD of a noise must be set to a number between 0 and 31. In general, the higher the number, the lower the pitch, with the exception of 0. Instead of producing the highest pitch possible, 0 produces the lowest pitch possible.

2. Type CALL ENVN and press RTN to generate the noise after you have defined it.

Experiment with defining different tones and noises. You can produce thousands of different sounds by changing different values.

NOTE: Your Master Component and your Computer Adaptor each contain a sound chip, with 3 channels on each chip. Channels 0-2 are on chip 1 and channels 3-5 are on chip 2. You can generate two different tone envelopes and two different noise envelopes at the same time — one tone envelope and one noise envelope on each chip. You can't generate a different envelope on each channel, because the computer always looks at the last envelope generated on each chip and uses that for all 3 channels on that chip. So if you generate a tone envelope through channel 2, it will be generated through channels 0, 1 and 2. This means, for example, that you cannot generate a different tone envelope for channel 0 and channel 2.

PUTTING IT ALL TOGETHER

Here is a short program that combines functions and routines to add color, sound and motion to an object. Before you type in this program, insert any Intellivision® game cartridge. Follow the directions on page 40 to 41, to display a moving object. Use the GRAB routine and number your object 0. (Use the equation O = 0.)

When you have displayed a moving object that you like, clear the screen. (Type CLR and press RTN.) Then type in the Animation Program exactly as it appears below. Remember to press RTN at the end of each statement.

When you have finished typing the program, clear the screen, then type RUN and press RTN. You should see the object you displayed change color as it moves horizontally across the screen to a musical scale accompaniment.

Animation Program

10 CALL HUSH

Start with no sound.

20 C = 0

Set channel. Tone will be generated through channel 0.

30 V = 15

Set volume. Tone will be generated at maximum volume.

40 SQ(0) = 40

Set speed at which object will move through its animation sequence. Moderate speed.

50 XV(0) = 10

Set speed at which object will move horizontally across the screen and start movement. Slow speed.

60 X = 0

Set up a variable that will represent the number code for a color. Give it a value of 0 (the number code for black)

70 FOR N = 12 TO 24

Set up a repeating loop that assigns values 12 through 24 to N (for use in the NOTE routine). N will receive a new value each time the loop is repeated.

80 CO(0) = X

Set the color of the object to the current value of X.

APPENDIX A

Dictionary of BASIC Terms

This section of the book cortains a brief description of the keywords, monitor commands, functions and routines in Intellivision BASIC. It is designed for the person who is already familiar with the BASIC language, and wants a quick reference guide to the Intellivision version of BASIC. If you are not familiar with BASIC, you may not understand everything in this section. Do send for a copy of the book, Step-By-S'ep Guide To Home Computing. (It's free.)

CATEGORY: BASIC KEYWORDS

CALL

This accesses an assembly language routine within the built-in BASIC or an Intellivisicn® cartridge. It allows you to do things that you cannot do using the BASIC language alone. (See page 36.)

Example:

```
CALL SHOW
```

Causes the SHOW routine to be executed.

CLR (Clear)

This clears the screen and returns the cursor to the home position. CLR can be used as a keyword in a program or as a command outside of a program.

Example:

```
10 CLR

20 PRIN "HELLO"
```

Clears the screen when the program is run, then displays the word Hello on the screen.

or

CLR (and press RTN)

Clears the screen when RTN *is pressed.*

DATA

This tells the computer to store numeric constants in a particular sequence, so they can be accessed at another point in the program by a READ statement. DATA statements must always have an associated READ statement.

Each numeric constant following the DATA keyword is separated from the others by a comma. The limit on data items in one DATA statement is 7 numeric constants.

Example:

10 FOR X=1 TO 3

Sets up a loop that will repeat 3 times.

20 READ A,B

Reads the numeric values in order from the DATA statement and stores these values as variables A and B.

30 PRIN A,B

Displays the current values of variables A and B.

40 NEXT X

Returns the computer to line 10 to repeat the loop.

50 DATA 5,10,15,20

Stores numeric values that will be read by the computer and assigned to numeric variables.

DIM (Dimension)

This is used to indicate the maximum number of elements in an array and to reserve memory space for the array.

The dimension of an array is set by writing the DIM keyword, followed by the array variable name, then the number of elements in the array in parentheses. The three array variable names that can be used are AA, AB and AC. The maximum number of elements in an array is 250, numbered from 1 to 250.

Example:

DIM AA(10)

Sets up an array named AA and reserves memory space for the array to contain 11 elements, numbered 1 to 11.

END

This keyword is used to mark the end of a program. When the computer reaches this statement during a program run, the run stops. The END statement is not always the last statement in a program.

Example:

```
10 PRIN "TYPE # FROM 5-10"
```

Displays the line in quotes.

```
20 INPU X
```

Stops program execution while numeric value for X is entered.

```
30 IF (X=7) GOTO 50
```

Looks to see if the value entered for X is 7. If it is, sends computer to line 50. If it is not, continues on with next statement.

```
40 GOTO 60
```

Sends computer to line 60.

```
50 END
```

Stops program execution

```
60 PRIN "TRY AGAIN"
```

Displays line in quotes.

```
70 GOTO 10
```

Sends computer back to line 10.

FOR...NEXT

These keywords are used to set up two "companion" statements which repeat any program steps between the two statements a specified number of times.

The FOR statement starts the loop, establishes a variable, gives it a starting value and makes a note of its ending value. Starting and ending values can be numeric constants or variables (in ascending order).

Each time the FOR statement is executed, it checks to see if the value of the variable is greater than the ending value established. f it is, the computer is sent to the statement after the NEXT statement. If it is not, the computer continues on with the statement(s) following the FOR statement.

The NEXT statement adds 1 to the current value of the variable and sends the computer back to its companion FOR statement.

A FOR...NEXT statement cannot be written and executed as a command.

Example:

10 FOR A = 1 TO 10

Sets up a locp that will repeat 10 times. Assigns a starting value of 1 to variable A and sets an ending value of 10.

20 PRIN A

Displays the current value of A.

30 NEXT A

Adds 1 tc the current value of A and sends the computer back to the FOR statement on line 1C.

GET

This temporarily halts program execution during a program run, so that a string entered from the keyboard can be assigned to a string varicble. The name under which the variable is to be stored is established by the GET statement.

Example:

50 GET A$

Halts program execution until a string is entered. Stores the string as variable A$.

60 PUT A$

Displays the value entered for A$.

GOTO (Go To)

This keyword causes the computer to "branch" to any specified program line, unconditionally.

If you write GOTO as a command, It acts as a RUN command, sending the computer to a specified program line and running the program from there.

Example:

10 GOTO 60

Sends computer ahead to line 60.

20 PRIN "PROVIDES FOR"

Displays line in quotes.

30 GOTO 80

Sends computer ahead to line 80.

40 PRIN "BRANCHING"

Displays the word in quotes.

50 END

Halts program execution.

60 PRIN "GOTO"

Displays the word in quotes.

70 GOTO 20

Sends computer back to line 20.

80 PRIN "UNCONDITIONAL"

Displays word in quotes.

90 GOTO 40

Sends computer back to line 40.

When program is run, computer displays: "GOTO PROVIDES FOR UNCONDITIONAL BRANCHING."

GSUB...RET

(Go Subroutine)... (Return)

Like GOTO, GSUB causes the computer to branch unconditionally to another point in the program, in this case to a subroutine. The first line number of the subroutine follows the GSUB keyword in the GSUB statement.

The subroutine ends in a RET statement which sends the computer back to the next statement following the GSUB statement from which it branched.

A subroutine can be called any number of times in a program. Up to 5 GSUB statements can be nested before a RETURN statement must be executed.

GSUB produces the same results as GOTO when executed as a command.

Example:

10 GSUB 40

Sends computer to subroutine beginning at line 40.

20 PRIN "END OF SUBROUTINE"

Displays line in quotes when computer returns from subroutine.

```
30 END
```

Stops program execution.

```
40 FOR X = 1 TO 3
```

Sets up a loop that will repeat 3 times.

```
50 PRIN "THIS IS A "
```

Displays line in quotes.

```
60 PRIN "SUBROUTINE"
```

Displays word in quotes.

```
70 NEXT X
```

Sends computer back to line 40 to repeat the loop

```
80 RET
```

Sends computer back to statement following GSUB statement from which it left (line 20).

IF

IF specifies a condition under which another program instruction will be executed. The condition is one in which two values are compared, using one of the comparison operators: = (equals), < (less than) or > (greater than). The values compared may be numeric variables, numeric constants or functions.

Examples: IF (A = 5)
IF (A > B)
IF (CO(3) = 7)

NOTE: You can also use the IF statement to compare individual characters from 2 different strings, if the strings are assigned to string variables. Example: IF (A$(1) = B$(2)) This compares the 1st character in the string assigned to A$ and the 2nd character in the string assigned to B$.

Example:

```
10 IF (A=B) GOTO 30
```

Looks to see if the value of A is equal to the value of B. If the two are equal, sends computer to line 30. If the two are not equal, continues to next statement.

INPU (Input)

This temporarily halts program execution during a program run, so that a numeric value, entered through the keyboard, can be assigned to a specific variable. The name under which the variable is stored is set by the INPU statement.

Example:

```
10 PRIN "HOW OLD ARE YOU"
```

Displays the line in quotes.

```
20 INPU A
```

Temporarily halts program execution while a numeric value is entered from the keyboard. Assigns that value to variable A.

```
30 PRIN "YOU ARE ",A
```

Displays line in quotes followed by value of A.

or

```
10 INPUT "ENTER NUMBER",A
```

Prints the string "ENTER NUMBER", then halts program execution while a numeric value is entered from the keyboard. Assigns that value to variable A.

PRIN (Print)

Tells the computer to display either a string enclosed in quotation marks, the result of an arithmetic operation (using either numeric constants or numeric variables) or the value of a numeric variable.

Example:

```
10 PRIN "HELLO"
```

Displays the word HELLO.

or

```
10 PRIN 5 + 6
```

Displays the result of adding 5 and 6 (11).

or

```
10 PRIN "THE NUMBER IS ",A
```

Displays the line in quotes, followed by the numeric value of A.

PUT

This causes the value of a string variable to be displayed.

Example:

```
10 SET A$ = "HELLO"
```

Assigns the string "HELLO" to string variable A$.

```
20 PUT A$
```

Displays the value of string variable A$.

READ

This is used to "read" items from a DATA statement and assign that data to a variable or variables, specified by the READ statement. Each time the READ statement is executed, the next item in the DATA statement (or block of statements) is read and a new value (the value of the data item being read) is assigned to the variable(s) in the READ statement.

If a READ statement does not assign all data items to variables, the next READ statement assigns following data items. If there are no more READ statements, the leftover data items are unused. If all data items are used before all READ statements are executed, the next READ statement gets the first data item again.

READ and DATA cannot be executed as commands.

Example: See DATA keyword for example.

REM (Remark)

REMark is used to insert comments or notes to yourself. It is often used to name a subroutine. The comments you insert with a REM statement are displayed when the program is listed, but do not appear when the program is run. During program run, the computer skips over REM statements.

REM cannot be executed as a command.

Example:

```
50 REM target subroutine
```

Identifies the subroutine when the program is listed.

SET

This is used to assign a value to a string variable when a program is being written. The value assigned must be a string, up to 20 characters long, enclosed in quotation marks.

Example:

```
10 SET A$ = "JUNK"
```

Assigns the string "JUNK" to string variable A$. Value is assigned when you write the program.

CATEGORY: MONITOR COMMANDS

CLOD (Cassette Load)

Loads a specific program into the computer's memory from an audio cassette tape. (See APPENDIX B.) A program is loaded by its assigned name.

Example:

```
CLOD GAME
```

Copies a program named GAME from cassette tape into computer memory.

CSAVE (Cassette Save)

Saves a program from computer memory to audio cassette tape, after the program has been given a name.

Example:

```
CSAVE GAME
```

Assigns the name GAME to a program and saves it from memory onto tape.

CVRF (Cassette Verify)

Verifies that a program saved on audio cassette matches the original program stored in memory. The program name must be given before the program can be verified.

Example:

```
CVRF GAME
```

Verifies that the program GAME stored on tape matches the program GAME stored in memory.

DEL (Delete)

This deletes a line or a group of lines from a program. (See page 28.) When deleting a single line, the word DEL is followed by the number of the line to be deleted. When deleting a group of lines, the word DEL is followed by the number of the first line in the group, a comma, then the number of the last line in the group.

Examples:

DEL 10

Deletes statement 10.

DEL 10,50

Deletes all statements from 10 to 50.

LIST

Allows you to print out a list of all statements in a program, a single statement in a program or a group of statements in a program. To list the entire program, type LIST alone. To list a single statement, tyoe LIST followed by the line number of the statement. To list a group of statements, type LIST, the line number for the first statement in the group, a comma, then the line number for the last statement in the group. Press the ESC key to stop listing before the program listing is completed.

Examples:

LIST

Lists entire program.

LIST 10

Lists statement 10 only.

LIST 10,60

Lists all statements from 10 to 60.

MENU

This gives a list of BASIC keywords, commands, functions, routines, cartridge functions and cartridge routines. There are six sub-menus. For a complete list of menu commands, see page 32.

Example:

MENU 1

Displays a list of all keywords used in Intellivision BASIC.

NEW

This erases a program from memory and resets any variables to zero. It does not erase moving objects or free the memory space that they are occupying.

Example:

Type NEW and press [RTN].

RUN

This tells the computer to execute a program that is stored in memory.

Example:

Type RUN and press [RTN].

CATEGORY: FUNCTIONS

A function is a way of representing a numeric value that has a special significance. A function is used in the same way as a variable, and is in fact a type of variable. Most of the functions below are used to manipulate and control moving objects. (For a further discussion of this particular use of functions, see page 36.)

The function NAME (2 letters) is followed by a NUMERIC VALUE in parentheses (either a number or a numeric variable) This tells the computer what the object or value the function relates to. It is called an ARGUMENT.

The function is set equal to a numeric value which gives the computer specific information about the argument. This numeric value can be changed to alter the nature of the argument in some way.

MOVING OBJECT FUNCTIONS:

CO (Color)

Sets the color for a moving object. The object number (from 0 to 7) follows the function name, in parentheses. The function is set equal to a number code for the color (a number from 0 to 15). Number codes for colors are as follows:

0 = Black
1 = Blue
2 = Red
3 = Tan
4 = Dark Green
5 = Green

6 = Yellow
7 = White
8 = Gray
9 = Cyan
10 = Orange

11 = Brown
12 = Pink
13 = Light Blue
14 = Yellow-Green
15 = Purple

Example:

CO(3) = 6

Sets color of object 3 to yellow.

or

10 A = CO(3)

Stores the number for the current color of object 3 as variable A.

```
20 PRIN A
```

Displays the number for the current color of object 3.

PC (Current Picture)

Use to display a particular picture in an animation sequence for a moving object. The argument is the moving object number. The function is set equal to the number of the picture's position in the sequence. This can be a number from 0 to 15. (There are 16 possible pictures in a motion sequence.)

Tells you the position number of the current picture being displayed in an animation sequence.

Examples:

```
PC(1) = 4
```

Displays the 5th picture in the animation sequence of object 1.

or

```
10 A = PC(1)
```

Stores the number of current picture being displayed, as variable A.

```
20 PRIN A
```

Displays the current picture number.

SQ (Sequence Speed)

Use to set the speed at which an object moves through its animation sequence. Argument is the object number. Function is set equal to any number from 0 (slowest) to 63 (fastest).

Displays the number for the current sequence speed at which a moving object is set.

Examples:

```
SQ(1) = 25
```

Sets the speed at which object 1 goes through its animation sequence at a moderate speed.

or

```
10 A = SQ(1)
```

Stores current sequence speed of object 1 in variable A.

```
20 PRIN A
```

Displays current sequence speed of object 1.

VS (Visibility)

Causes a moving object to appear or disappear. Argument is object number. Function can be set equal to 1 or −1. −1 causes object to disappear. 1 causes object to reappear.

Displays the number for the current visibility status of an object.

Examples:

```
VS(1) = -1
```

Causes moving object 1 to disappear.

or

```
10 A = VS(1)
```

Stores number for current visibility status of object 1 as variable A.

```
20 PRIN A
```

Displays number for current visibility status of object A.

XM (Horizontal Mirror)

Turns a moving object so that it faces the opposite direction. Argument is object number. Function can be set equal to 1 or −1. 1 turns object in opposite direction. −1 returns object to original direction.

Displays the number for the current horizontal mirror status of an object.

Examples:

```
XM(1) = 1
```

Turns moving object 1 to face the opposite direction.

or

```
10 A = XM(1)
```

Stores the current horizontal mirror status of object 1, in variable A.

```
20 PRIN A
```

Displays the number for the current horizontal mirror status of object 1.

XP (Horizontal Position)

Sets the horizontal screen position of a moving object. Argument is object number. Function can be set equal to any numeric value from 0 (off the left side of the screen) to 167 (off the right side of the screen).

Displays the number for an object's current horizontal screen position.

Examples:

```
XP(1) = 90
```

Moves object 1 slightly right of center screen.

or

```
10 A = XP(1)
```

Stores the number for object 1's current horizontal screen position, as variable A.

```
20 PRIN A
```

Display the number for object 1's horizontal screen position.

XS (Object Width)

Doubles the width of a moving object. Argument is object number. Function can be set equal to 1 or −1. 1 doubles the object width. −1 returns object to original shape.

Displays the number for the current width of an object.

Examples:

```
XS(1) = 1
```

Doubles the width of object 1.

or

```
10 A = XS(1)
```

Stores the number for the current width of object 1, as variable A.

```
20 PRIN A
```

Displays the number for the width of object 1.

YV (Vertical Velocity)

Sets the velocity at which an object moves vertically across the screen. Starts the object in motion. Argument is object number. Function can be set equal to any number from −127 to 127. Negative numbers move object from bottom to top of screen. Positive numbers move object from top to bottom of screen.

Displays number for current vertical velocity of a moving object.

To stop vertical movement, set function equal to 0.

Examples:

YV(1) = −20

Starts object 1 moving vertically up the screen, at slow speed.

or

10 A = YV(1)

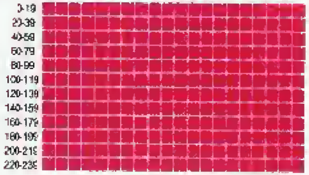
Stores current vertical velocity of object 1, as variable A.

20 PRIN A

Displays vertical velocity of object 1.

OTHER FUNCTIONS:

BK (Background Color)



Sets the background color of a specific rectangle, or "card", on the screen. The screen is divided into 240 cards on a 12 by 20 grid, number from 0 to 239.

Argument of the function is the card number. Function can be set equal to any of the 16 color numbers (from 0 to 15). See color list on page 58.

Displays the number for the background color of a particular card on the screen.

Examples:

10 X = 0

Assigns a value of 0 to X.

20 FOR A = 0 TO 239

Sets up a loop that repeats 240 times, with a new value assigned to A each time the loop repeats, starting at 0 and ending with 239.

```
30 BK(A) = X
```

Sets the background color of "card" A (the current number value of A) to the current numeric value of X, starting with 0 (black).

```
40 X = X+1
```

Increases the value of X by 1 each time the loop is executed.

```
50 IF (X=3) X=0
```

Resets X to 0 when it reaches a value of 3, thus restricting the background colors displayed to 0 (black), 1 (blue) and 2 (red).

```
60 NEXT A
```

Returns the computer to line 20.

FM (Format)

Allows you to format the manner in which numeric data will be displayed on the screen or printed on a printer.

The argument of this function can be a number from 0 to 4. The argument establishes how many numbers will be displayed on a line, the color in which they will be displayed and whether they will be displayed in decimal or scientific notation.

FM(0) sets format of 2 decimal numbers per line, right justified. The number to which the function is set equal determines the number of digits right of the decimal. (Maximum 7 digits right of decimal.)

Example:

```
A = 10.7536
```

Assigns a value to variable A.

```
B = 20.8424
```

Assigns a value to variable B.

```
FM(0)=3
```

Sets format for 2 numbers per line, with 3 digits right of decimal point.

```
PRIN A,B
```

Displays.

```
     10.754          20.842
```

FM(1) sets format of 1 decimal number per line, right justified. The number to which the function is set equal determines the number of digits displayed to the right of the decimal. (Maximum 7 digits right of decimal.)

Example:

```
FM(1)=2
```

Sets format for 1 number per line, with 2 digits right of decimal point.

```
PRIN 7.673
```

Displays:

```
                    7.67
```

FM(2) sets format of 2 numbers per line, displayed in scientific notation. The function can be set equal to any number — the number displayed in scientific notation will always have 3 digits left of the E and 2 digits right of the decimal point.

Example:

```
10 A = 936000
```

Assigns a value to variable A.

```
20 B = 22500
```

Assigns a value to variable B.

```
30 FM(2)=0
```

Sets format for 2 numbers per line, expressed in scientific notation.

```
40 PRIN A,B
```

Displays:

```
    9.36E5        2.25E4
```

FM(3) sets format of 1 number per line, displayed in scientific notation. The function can be set equal to any number. See FM(2).

Example:

```
10 X = 4320
```

Assigns a value to variable X.

```
20 FM(3)=7
```

Sets format for 1 number per line, expressed in scientific notation.

```
30 PRIN X
```

Displays:

4.32E3

FM(4) sets the color in which characters will be displayed. Function may be set equal to a number between 0 and 7. (See color code list on page 58.) If not set, the function defaults to FM(4)=0.

Example:

```
FM(4)=7
```

Sets color of all characters displayed to white.

```
PRIN 1
```

Displays the character 1 in white.

IT (Integer)

This deletes ("truncates") all numbers right of the decimal point in a decimal number. The argument for the function is the decimal number. This number must be within the range 32767 to −32767.

Example:

```
A = IT(1.23)
```

Changes all digits right of the decimal point to 0 for the number 1.23. Stores the resulting integer as variable A.

```
PRIN A
```

Displays: 1.00

RN (Random Number)

Generates a randomly selected number from 0 to 99 The argument for the function is a "dummy" argument, which means that it can be any number. (0 is commonly used.)

Example:

```
A = RN(0)
```

Selects an integer from 0 to 99 at random and stores it as variable A.

```
PRIN A
```

Displays the random number selected.

CATEGORY: ROUTINES

DIST (Distance)

Returns the distance between two moving objects and stores that distance in the variable D (which can be displayed with a PRIN statement). Requires that the number of one object be assigned to N and the number of the other object be assigned to M. If objects occupy different horizontal and vertical positions, the DIST routine will return the SUM of the horizontal and vertical distances between them.

Example:

```
YP(0) = 10
```

Sets the horizontal position of object 0 at 10 (left side of screen).

```
YP(0) = 10
```

Sets the vertical position of object 0 at 10 (top of screen).

```
XP(1) = 10
```

Sets horizontal position of object 1 at 10 (same as object 0).

```
YP(1) = 20
```

Sets vertical position of object 1 at 20 (10 units lower than object 0).

```
N = 0
```

Stores object number 0 as variable N.

```
M = 1
```

Stores object number 1 as variable M.

```
CALL DIST
```

Returns the distance between object 0 and object 1.

```
PRIN D
```

Displays the distance between object 0 and object 1. (Result: 10.00)

ENVN (Envelope Noise)

Generates a "noise" with variable envelope, period, length and channel, for which values must be set. (See envelope description on page 42.) Range of values for each variable are as follows:

C (channel): 0 to 5
P (period): 0 to 31 (1 produces highest pitch, 31 and 0 produce lowest pitch.)
L (length): 0 to 65,000
E (envelope): 0 to 15 (see below)

Envelope values (other numbers between 0 and 15 not listed duplicate effects noted below):

0 Decay only. Sound drops from peak & stops.
4 Attack only. Sound rises to peak & stops.
8 Decay only repeats indefinitely.
10 Alternates indefinitely between attack and decay, starting with a decay.
11 Initial decay, followed by a sustain of peak volume.
12 Attack only repeats indefinitely.
13 Initial atack, followed by sustain of peak volume.
14 Alternates indefinitely between attack and decay, starting with attack.

Example:

10 C=0

Selects channel 0 for noise to be generated through.

20 P=11

Selects a mid-range pitch for the noise.

30 L=10000

Sets the length of the envelope.

40 E=0

Sets envelope for decay only.

50 CALL ENVN

Generates noise.

ENVT (Envelope Tone)

Same as ENVN, except tone is generated instead of noise and range of values for period is 0 to 4095, with 0 procucing the highest pitch and 4095 the lowest pitch.

Example:

10 C=3

Selects channel 3 for tone to be generated through.

20 P=200

Selects a high pitch.

30 L = 10000

Sets the length of the envelope.

40 E = 0

Sets envelope for decay only.

50 CALL ENVT

Generates the tone.

GRAB (Grab)

Displays and stores a defined moving object in memory, for use in programs. Object definition requires that values be established for object, picture, number of pictures in animation sequence, and resolution. (See discussion on page 40.) Range of possible values:

O (object): 0 to 7
N (picture): 0 to 127
M (number of pictures in sequence): 0 to 15
D (resolution) 0 to 2

Temporarily uses variables W, X, Y and Z.

Example:

10 O = 1

Assigns 1 as the moving object number.

20 N = 2

Selects picture 2 for moving object 1.

30 M = 5

Sets an animation sequence of 6 pictures, starting with picture 2.

40 D = 2

Set double resolution for moving object 1, so both top and bottom will be displayed.

50 CALL GRAB

Displays the moving object and stores it in memory.

Use of the GRAB routine requires that an Intellivision® game cartridge be inserted in the cartridge port.

HAND (Hand Controller)

Displays number representing last control pressed on Hand Controller. Stores this value as a variable. Number values for controls are:

Disc: −1 to −16 (starting at −1 and decreasing clockwise) if Disc is being pressed when HAND is called.

−101 to −116 (starting at −101 and decreasing clockwise) if Disc has been released when HAND is called.

NUMBER VALUES FOR THE DISC ARE STORED AS VARIABLE H.

Keypad: 0 to 11 (Clear is 10 and Enter is 11) if key is being pressed when HAND is called 100 to 110 if key has been released when HAND is called.

NUMBER VALUES FOR THE KEYPAD ARE STORED AS VARIABLE H.

Action Buttons: 0 — No button being pressed when HAND is called. 1 — either top button being pressec. 2 — lower left button being pressed. 3 — lower right button being pressed.

NUMBER VALUES FOR THE ACTION BUTTONS ARE STORED AS VARIABLE A.

Example:

```
10 CALL HAND
```

Checks the last control pressed on a Hand Controller and stores the numeric value for that position as variable H or A.

```
20 IF (H > 0) GOTO 10
```

If any key on keypad is being pressed, ignores that key and returns to line 10.

```
30 IF (H <  −99) GOTO 10
```

If Disc has been released, ignores value of H and returns to line 10.

```
40 XP(0) = −H * 10
```

If Disc is currently being pressed, changes value of H to positive number, multiplies it by 10 and uses it to set the horizontal position of object 0.

```
50 GOTO 10
```

Sends the computer back to line 10 to repeat the loop.

HUSH (Stop Sound)

This routine kills all sound generated by other routines (tone or noise).

To stop the sound on one channel only, set the volume or envelope values on that channel to zero.

Example:

CALL HUSH

Stops generation of tone or noise.

LINK

Gives two objects the same velocity and maintains a constant distance between them. Velocity set for the SMALLER object number determines the velocity at which both objects will move.

Requires that O be set equal to the LARGER object number of the two objects being moved.

Horizontal and vertical velocities set for the LARGER object number does not affect velocity, but instead sets the horizontal and vertical distance that will be maintained between the two objects, as they move.

Example:

10 O=1

Assigns 1 as the moving object number. Objects 0 and 1 will be linked.

20 CALL

Links objects 0 and 1.

30 XV(1) = 10

Sets the horizontal distance to be maintained between objects 0 and 1

40 YV(1) = 5

Sets the vertical distance to be maintained between objects 0 and 1.

50 XV(0) = 10

Sets the horizontal velocity of both objects at slow speed and moves objects across the screen, from left to right.

NOIS (Noise)

Generates a "noise" sound with variable channel, period and volume. Range of values that can be assigned to variables are as follows:

C (channel): 0 to 5
P (period): 0 to 31 (1 produces highest pitch while C & 31
produce lowest pitch)
V (volume): 0 to 15

Example:

```
10 C =
```

Selects channel 2 for noise to be generated through.

```
20 P = 18
```

Selects mid-range pitch.

```
30 V = 15
```

Selects peak volume.

```
40 CALL NOIS
```

Generates noise.

To stop noise, type CALL HUSH and press RTN

NOTE

Sets the period of a tone to be generated to a particular note on a musical scale that ranges from 0 (low note) to 95 (high note). Does not generate a tone, unless used with the TONE routine. Requires that a value from 0 to 95 be set for N (note).

Example:

```
10 N = 20
```

Selects note 20 from musical scale of 0 to 95.

```
20 CALL NOTE
```

Sets value of P (period) to that of note 20.

```
30 CALL TONE
```

Generates tone with period set by NOTE routine.

OUTP (Output)

Directs listings of programs either to the TV screen or to a printer, depending on the value assigned to D. When D = 1, program lists on the screen. When D = −1, program lists on printer. If no value is set for D, program lists on screen by default.

EXAMPLE:

```
D = −1
CALL OUTP
LIST
```

Instructs computer to PRINT the listing of the program *stored in memory, rather than displaying it on the TV screen.*

Use of the OUTP routine requires that a printer be hooked up to your Computer Adaptor.

SHOW

Displays a moving object on screen without permanently storing it in memory. Does not allow object to be animated or used in a program. Temporarily uses variables U, V, W, X, Y and Z. Requires that values be set for object, picture and resolution. (See GRAB routine for range of values.)

Example:

O = 2

Assigns 2 as the moving object number.

N = 0

Selects picture 0 for moving object 2.

D = 1

Selects single resolution, so only top OR bottom half of object will be displayed.

CALL SHOW

Displays moving object, but does not store it in memory.

Use of the SHOW routine requires that an Intellivision® cartridge be inserted in the cartridge port.

TONE

Generates a tone over which you control period (which determines pitch), volume, and the channel through which the tone will be generated.

Range of values that can be assigned to variables are as follows:

C (channel) = 0 to 5
V (volume) = 0 to 15
P (period) = 0 to 4095 (0 produces the highest pitch and 4095 produces the lowest pitch)

Example:

C = 0

Selects channel 0 for tone to be generated through.

V = 15

Selects maximum volume.

P = 200

Selects high pitch.

CALL TONE

Generates the tone.

To stop tone, type CALL HUSH and press RTN.

CONSTANTS AND VARIABLES

A constant is a value — numeric or string — that does not change.

A variable is a value — numeric or string — that can change.

Example:

A = 10

A is a numeric variable. 10 is a numeric constant.

Example:

A$ = "Hello"

A$ is a string variable. "Hello" is a string constant.

Numeric variables can be labeled with any one of the letters of the alphabet.

There are 3 string variable labels: A$, B$ and C$. This means a maximum of 3 string variables per program. Each string variable can be assigned a value up to 20 characters long.

CONSTANTS & VARIABLES IN PROGRAM STATEMENTS

PRIN may be followed by:

■ ONE string constant up to 20 characters long, that immediately follows the keyword. Example: PRIN "HELLO"

and/or

■ ONE or MORE numeric constants, variables (up to 7) or arithmetic expressions, separated from each other by commas.

Examples:

PRIN A,B,C,D

PRIN "THE NUMBER IS ", A

PRIN "THE ANSWER IS ", A + B

INPU may be followed by:

- ONE or MORE numeric variables (up to 7)

Example: INPU A, B, C

or

- ONE string constant up to 20 characters long, immediately following the keyword, with ONE or MORE numeric variables, separated by commas, after the string.

Example: INPU "THE NUMBERS ARE ", A, B

GET
PUT may be followed by ONE string variable.

Example: PUT A$

Assign Statements:

- To assign a value to a NUMERIC VARIABLE when you write a program, use a simple assign statement.

Example: 10 A=5

- To assign a value to a STRING VARIABLE when you write a program, use a SET statement.

Example: 10 SET A$="HELLO"

Variable values may be assigned and changed using commands instead of program statements. This allows you to save the memory space required for the line numbers of assign statements.

Example:

A=5

B=7

10 PRIN "A + B = ", A+B

ARRAYS

An array is a GROUP OF VALUES assigned to a variable. Each individual value in the group occupies a "slot" in the array. The number of slots is established by a DIMension statement. An individual value (called an element) can be used in a program statement in exactly the same way as any other value.

Only NUMERIC arrays are possible in Intellivision® BASIC. There are 3 numeric arrays AA, AB and AC. This means a maximum of 3 numeric arrays per program.

Each array may be dimensioned with up to 250 elements (numbered 1 to 250).

PUNCTUATION

COMMAS are used to separate constants, variables or data items from each other when these are used in a command or statement.

Example:

```
PRIN A,B,C

10 DATA 10,20,30,40

50 PRIN "THE ANSWER IS ",X
```

QUOTATION MARKS are used to define a literal string.

Example:

```
PRIN "INTELLIVISION"

10 PRIN "THE SUM OF A + B IS ", A+B
```

PARENTHESES are used to enclose the argument of a function or the condition of an IF statement.

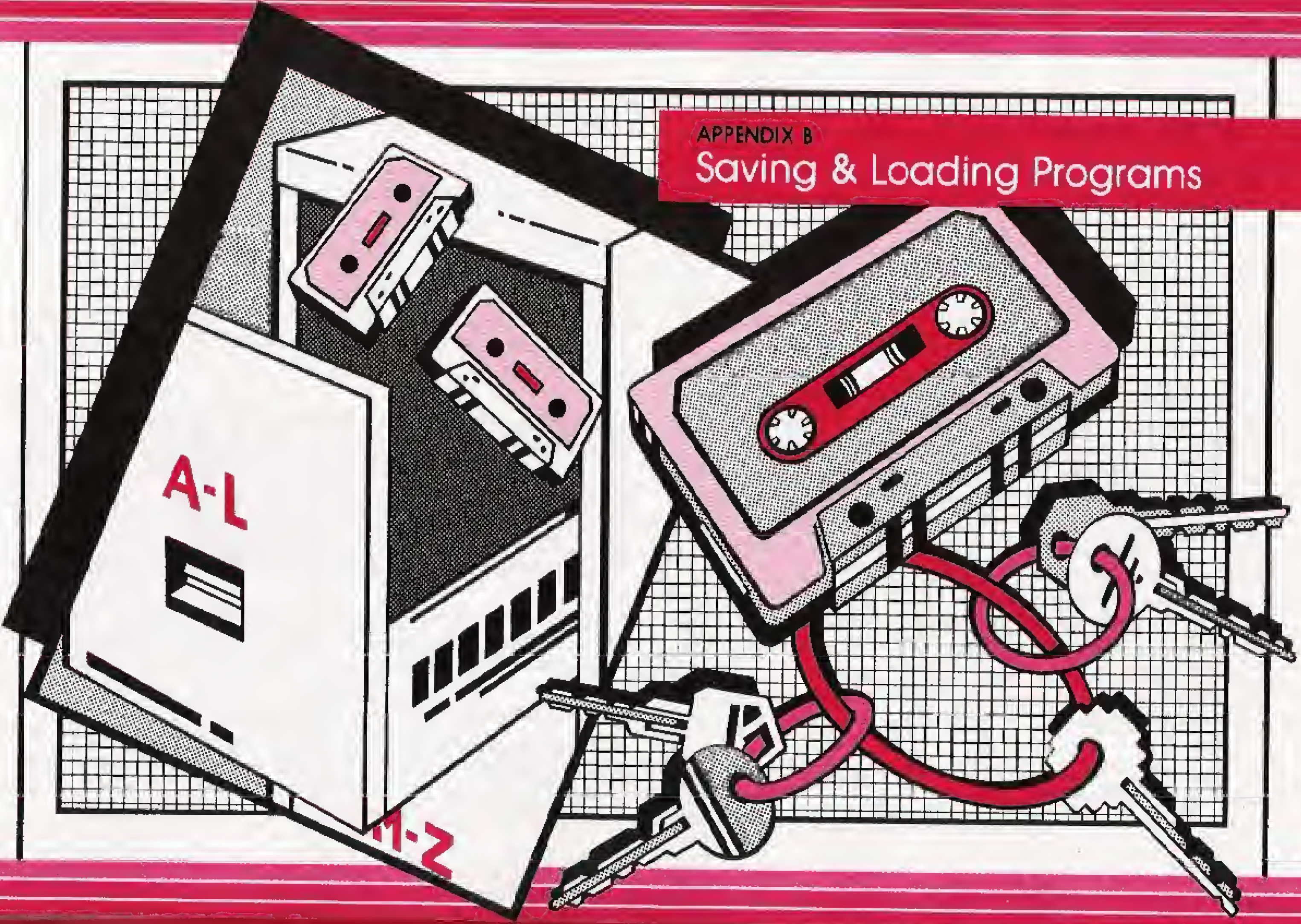
Example:

```
CO(2) = 5

30 IF (A < 10) GOTO 90
```

APPENDIX B

Saving & Loading Programs



Three BASIC commands control cassette recordings of your programs. These are:

CSAV...This copies the BASIC program from memory to cassette.

CLOD.. This loads a program from cassette tape back into memory.

CVRF...This compares a program saved on tape to the program stored in memory and verifies that they match.

Each of these commands is followed by a program name, up to 4 characters long. Example: CSAV PROG (The use of a name is optional, but if you don't specify a name, it is difficult to load a program without knowing EXACTLY where it is located on tape.)

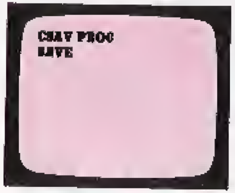
With or without a name, the steps for saving, loading or verifying a program are as follows:

1. Type the command CSAV, CLOD or CVRF, followed by the program name. Then press RTN. The computer will display the word SET. (When you verify or load a program, be sure to type exactly the same name that you used to save the program.)

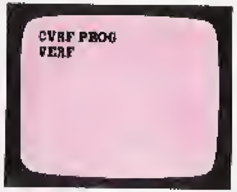
2. Use the FAST FORWARD or REWIND keys on your cassette recorder to position the tape at the point where you plan to save your program, or at the beginning of the program you plan to load. (Always keep an accurate written record of where your programs start and end on the tape counter.)

3. After the tape is positioned, press ANY KEY OR THE DISC ON EITHER HAND CONTROLLER. The computer will display the word GO.

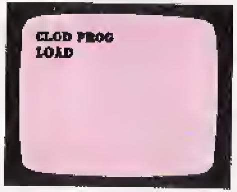
4. Now press PLAY (to load or verify) or PLAY AND RECORD (to save) on the cassette recorder. The tape begins when you press ANY KEY OR THE DISC ON THE HAND CONTROLLER.



```
CSAV PROG
SAVE
```



```
CVRF PROG
VERF
```



```
CLOD PROG
LOAD
```

One of the above messages shows on the screen, depending on which command you used. The save is completed when the cursor shows again.

When LOADING, the computer first displays: LOAD PROG. This tells you which program the computer is searching for. When a program is found, the computer displays the name of the found program. If the found program matches the program being searched for, the computer loads the found program into memory. If the two names do not match, the computer continues searching, until it finds the correct program or you press the ESC key. If you have missed the correct starting point on tape for your program, the computer will not be able to find it. Rewind the tape and try again. When the load procedure is completed, the cursor will appear again.

When VERIFYING, the computer reads every program it encounters on the tape and displays the program name to the right of the VERF message. When it reads a program, it checks two things:

■ the program name

■ the program data

If NEITHER the program name nor the program data found matches the program in memory, the program name is colored gray. The computer continues searching for the matching program.

If the program name DOES NOT MATCH, but the program data DOES, the program name remains colored black on a green background, and the search continues.

If the program name DOES MATCH, the search stops, the cursor appears and the computer checks the program data.

■ If the program data matches, the VERF message remains colored yellow on green. The program name remains black on green.

■ If the program data does not match, the VERF message and the program name are colored gray.

NOTES: After a CSAV, CLOD or CVRF command is entered, its execution can be halted by pressing the ESC key.

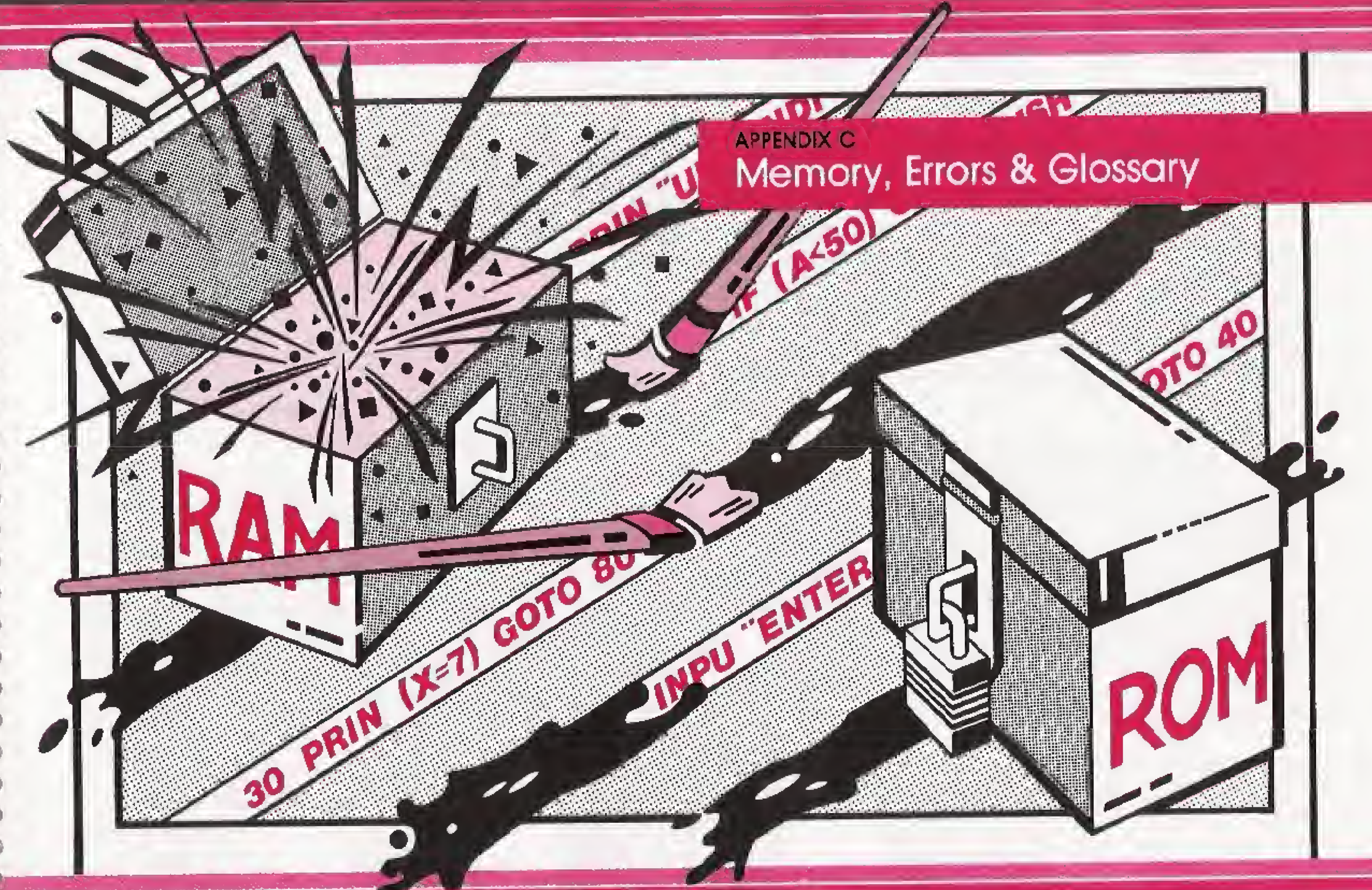
Always use cassette tapes no longer than 30-60 minutes in length (C30 or C60 tapes). Remember to move your tape beyond its plastic leader when saving or loading or you will lose data.

Keep an accurate written record of your programs, including name and start and end positions on the tape. Use the tape counter on your cassette recorder to get accurate start and end positions.

Try to put no more than one or two programs per side on a cassette tape. Save the same program two or three times in a row. This assures you of an accurate save and reduces the chances of missing your program when loading. In addition, always make a backup (duplicate) tape of programs you plan to save for a long time.

DO NOT place your cassette tapes on top of the TV set, in front of the screen or near any other electrical devices which generate a strong magnetic field.

APPENDIX C
Memory, Errors & Glossary



MEMORY

Your computer uses two kinds of memory — ROM and RAM.

ROM stands for Read Only Memory and is the computer's permanent memory. You cannot erase, change or add to data stored in ROM. You can only tell the computer to read and use this data. Most of the computer's internal operating instructions are stored in ROM, as well as the built-in BASIC and the "interpreter" that translates BASIC into the assembly language that the computer understands. Your Intellivision® games are also stored in ROM on your game cartridges

RAM stands for Random Access Memory This is your computer's "working" memory, the memory that you can use. You temporarily store data in RAM when you write a program or load a program from cassette tape. But RAM is not permanent. You erase everything stored in RAM when you turn your Master Component off or press the RESET button.

The length of program you can write is determined by the amount of RAM you have available. Your Computer Module has a maximum of 2048 bytes of RAM available. A BYTE is the amount of memory needed to store one character or space that you type.

If you run out of memory in the middle of writing a program or command (say, to display a moving object), the entire line at which you ran out of memory will turn gray when you press RTN.

ERRORS

The computer responds to different types of errors in different ways. Some errors show up as soon as you enter a command or program statement. Others don't show up until you run your program.

When you enter a statement or command, the computer color codes everything that it understands. If it understands enough to execute the statement or command, it leaves the line as is. If it does not understand enough to execute the statement or command, it re-colors everything that it understood gray (leaving the rest uncolored).

IMPORTANT: If you type a statement or command incorrectly, the computer may recognize enough to execute it IN A DIFFERENT WAY THAN YOU INTENDED. For example, if you type PRIN A$, the computer will color PRIN A. It will then look for a NUMERIC value of A. If it does not find one, it will assume a value of 0 and print 0.00. It will NOT print the STRING value of A$, since you must use the PUT keyword to print a string variable.

If you LIST a program containing non-executable statements, those statements will be displayed in white characters.

If you RUN a program containing non-executable statements, those statements will be listed in white characters before the program starts. Non-executable statements will then be skipped over during the program run.

Up to five subrcurines can be nested before a RETURN statement must be executea. Any nested subroutines beyond the maximum are listed in black during the program run, but otherwise ignored. (Nested subroutines are explained in the Step-By-Step Guide To Home Computing book.)

Attempts to divide by zero are refused with a short whistle from your TV speaker.

If you attempt to enter a command or statement with insufficient memory, the entire line is displayed in gray when you press RTN. Attempts to display a moving object using a GRAB routine result in a long whistle.

If you try to use the SHOW or GRAB routines without a cartridge inserted, you hear a long whistle.

COLOR CODES FOR BASIC PROGRAM ELEMENTS

Each of the elements of a program turns a specific color when enterec with the RTN key.

Element	Color Code Characters/Background
LINE NUMBERS (at the beginning of a line or following a GOTO or GSUB keyword)	Tan/Green
BASIC KEYWORDS (PRIN, GOTC, etc.)	Black/Pink
LITERAL STRINGS (characters enclosed in quotation marks. Example SET A$ = "Hello" or PRIN "10 + 17")	Yellow/Blue
CONSTANTS (six or fewer numbers with or without a decimal point)	Black/Light Blue
NUMERIC VARIABLES (a single alphabetic character that represents a numeric value)	Black/Yellow

Element — Color Code Characters/Background

STRING VARIABLES . . . . Blue/Yellow

(A$, B$ and C$, used to represent a literal string)

FUNCTIONS . . . . White/Black

(the 2-character function NAME only, including array variable names. Function argument is color coded as a numeric constant or variable, whichever is used. (Exolanation of functions on page 58.))

ROUTINE NAMES . . . . White/Blue

(SHOW, GRA3, TONE, ENVN, etc.)

TEXT . . . . Yellow/Brown

(any characters following a REM keyword)

ARITHMETIC OPERATORS . . . . Green/White

(+ , − , * , / and parentheses)

RELATIONAL OPERATORS . . . . Black/Tan

(=, < or > used to compare two values)

DELIMITERS . . . . Blue/White

(any character used to tell BASIC where something starts or ends. Includes commas, quotation marks, parentheses and sometimes the = sign or the word "to".

```
Example: FOR A = 1 TO 3
         SET A$ = "HELLO")
```

MONITOR COMMANDS (RUN, LIST, NEW, etc.) are displayed in black characters against the standard green background, just as typed.

When you LIST a program, it is displayed in black characters against the stancard green background.

SPECIAL EDITING TECHNIQUES

The use of ARROW KEYS to move the cursor and correct errors is described on page 22. These same keys can be used in even more specific ways to make program writing easier.

RE-EXECUTE A COMMAND. Move the cursor over the first letter of the command, type that letter, then press the UP or DOWN ARROW key instead of RTN.

REPEAT A PROGRAM LINE. Move the cursor over the first character of the line number. Type a new line number and press the UP or DOWN ARROW key instead of RTN. The entire line will now appear twice when listed — once with the old line number and again with the new line number.

REPEAT A PROGRAM LINE WITH SOME ELEMENT CHANGED. Move the cursor over the first character of the line number. Type a new line number. Use the RIGHT ARROW key to move the cursor right to the character(s) you want changed. Type the new characters. Then press the UP or DOWN ARROW key instead of RTN. When you list the program, the line will appear in its original form with the old line number and in its altered form with the new line number.

Example: To type 10 X = 0
                              20 Y = 0

Position the cursor over the 1 in 10 and type 20. Move the cursor right to the X in line 10 and type Y. Then press the UP or DOWN ARROW key instead of RTN.

TO TRUNCATE PART OF A LINE. Move the cursor to the last character which you wish to include in the line. Type that character and press the UP or DOWN ARROW key instead of RTN. Everything to the right of the character you typed will be deleted.

GLOSSARY OF TERMS

ARRAY. A group of values numbered and stored in a particular order under a single variable label.

ARRAY ELEMENT. An individual value in an array.

ARGUMENT. A numeric value enclosed in parentheses, which represents the object or value on which a function will operate.

ASSEMBLY LANGUAGE. The math-based language which the computer directly understands without an "interpreter".

BASIC. A "high level" programming language, based on the English language. Stands for Beginner's All-Purpose Symbolic Instruction Code.

BRANCH. To divert program execution to an alternate point in the program and continue execution from there.

BYTE. The amount of memory space needed to store a single typed character.

COMMAND. An instruction written without a line number and executed immediately after RTN is pressed.

CONSTANT. A value that does not change.

CURSOR. The small square that indicates the next screen position in which a character will appear when typed.

DATA. Another word for information.

DIMENSION. The number of individual values (elements) in an array.

EXECUTE. To carry out, as to "execute a command".

FUNCTION. A way of representing a number value that has a special significance in regard to an object or another value. A special kind of variable.

HARDWARE. The actual mechanical, magnetic and electronic structure of a computer and its peripherals.

HOME POSITION. The upper left corner of the screen.

INCREMENT. (noun) The amount by which a value increases. (verb) To increase in value.

INTEGER. A whole number.

KEYWORD. A BASIC word that has a special meaning when used in a command or statement. The "building blocks" of a program.

LOOP. A repeating group of statements.

MONITOR COMMAND. In general, a command that tells the computer to do something with a program that has been written or loaded into memory.

MOVING OBJECT. Any of the objects which appear in a game stored on an Intellivision® game cartridge; i.e., baseball players, robots, aliens, submarines, etc.

OUTPUT. Data that the computer gives you in any form (displayed on a TV screen, printed on paper or stored on cassette tape).

PERIPHERAL. A hardware device that connects to your computer to extend its capabilities.

PROGRAM. A list of numbered instructions (statements) that are stored until a RUN command is given, then executed in order.

RAM (Random Access Memory). The "working memory" in which data is temporarily stored. Data in RAM can be edited. Data is erased when power to the computer is turned off.

ROM (Read Only Memory). Memory in which data is permanently stored. Data in ROM cannot be edited. Data is not erased when power to the computer is turned off.

ROUTINE. A "mini-program" written in assembly language and stored in ROM, which can be accessed by a BASIC command or statement. Routines allow you to do things which are not possible using the BASIC language alone.

SCIENTIFIC NOTATION. A method of expressing very large or very small numbers.

SOFTWARE. The externally stored programs with which a computer works.

STATEMENT. A numbered instruction in a program.

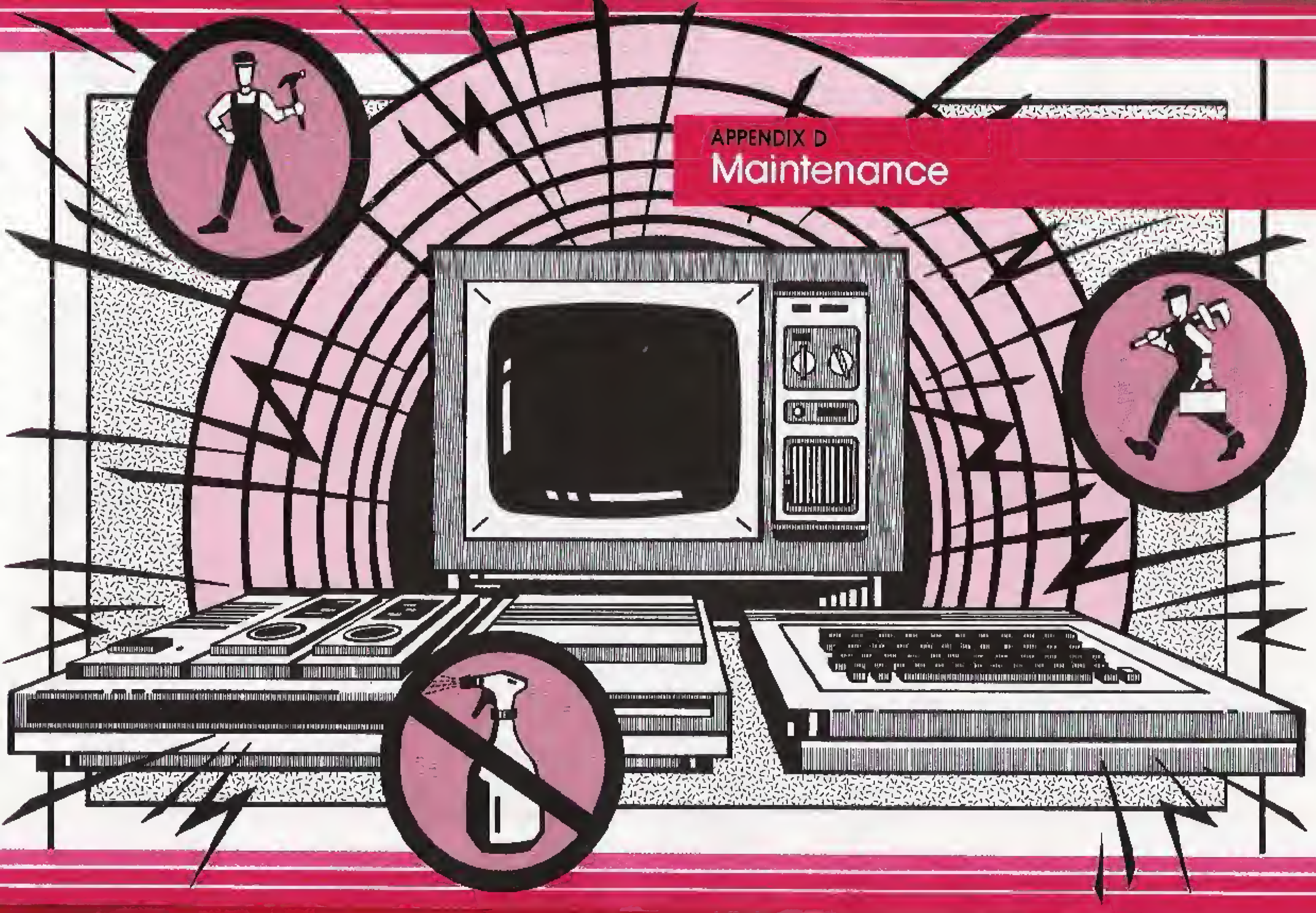
STRING. A group of characters enclosed in quotation marks.

TRUNCATE. To shorten a number or string by dropping digits or characters, starting from the right.

VARIABLE. A value that can change.

APPENDIX D

Maintenance



- Clean the outer surface of your module with a clean cloth, either dry, dust-sensitive or only slightly dampened in mild, soapy water.

- NEVER use solvents or harsh detergents of any kind to clean the surface of the Computer Module. NEVER spray ANY kind of liquid on or near the Computer Module.

- Be particularly careful of moisture near the keyboard. Clean the keys with a DRY cloth or DRY soft brush ONLY.

- Never open the chassis of your Computer Adaptor or Computer Keyboard. For service information, call one of these numbers:

SERVICE/INFORMATION NUMBERS

Eastern U.S 1-800-257-5185

New Jersey residents, call direct or collect: 1-609-655-3533

Western U.S. 1-800-421-2826

California, Alaska & Hawaii residents, call direct or collect:
1-213-978-6850

Warranty

COMPUTER ADAPTER

Mattel Electronics warrants to the original consumer purchaser of its Intellivision® COMPUTER ADAPTER while in the United States that the product will be free of defects in material or workmanship for 90 days from the date of purchase under normal in-house use.

During the warranty period Mattel Electronics will at its option, repair or replace the product without charge for parts or labor, when returned postage prepaid and insured to a Mattel Electronics authorized service center with proof of date of purchase. Mattel Electronics reserves the right to utilize reconditioned parts in repairing the product or to utilize reconditioned units in replacing the product.

This warranty excludes incidental or consequential damages resulting from the product or use of the product (some states do not allow the exclusion of incidental or consequential damages, so the foregoing exclusion may not apply to you).

Please read the Owner's Manual carefully before using the product. In the unlikely event you do experience difficulty, please call one of the Mattel Electronics Service/Information numbers shown below.

Units returned without proof of the date of purchase or units returned after the 90 day warranty period has expired, will be repaired or replaced (at our option) for a service charge. Call one of the following telephone numbers to obtain the location of the nearest service center and the amount of the repair charge.

In the **EASTERN UNITED STATES (800) 257-5185;**

NEW JERSEY RESIDENTS MUST CALL (609) 655-3533 direct or collect;

in the **WESTERN UNITED STATES (800) 421-2826;**

ALASKA, CALIFORNIA, or **HAWAII RESIDENTS MUST CALL (213) 978-6850** direct or collect.

ATTENTION MILITARY PERSONNEL, if your Intellivision® COMPUTER ADAPTER requires service while on duty in a foreign country, you should contact the military exchange serving your area.

IF YOU NEED SERVICE, CALL ONE OF THE SERVICE/INFORMATION NUMBERS ABOVE.

COMPUTER KEYBOARD

Mattel Electronics warrants to the original consumer purchaser of its Intellivision® COMPUTER KEYBOARD while in the United States that the product will be free of defects in material or workmanship for 90 days from the date of purchase under normal in-house use.

During the warranty period Mattel Electronics will at its option, repair or replace the product without charge for parts or labor, when returned postage prepaid and insured to a Mattel Electronics authorized service center with proof of date of purchase. Mattel Electronics reserves the right to utilize reconditioned parts in repairing the product or to utilize reconditioned units in replacing the product.

This warranty excludes incidental or consequential damages resulting from the product or use of the product (some states do not allow the exclusion of incidental or consequential damages, so the foregoing exclusion may not apply to you).

Please read the Owner's Manual carefully before using the product. In the unlikely event you do experience difficulty, please call one of the Mattel Electronics Service/Information numbers shown below.

This warranty does not apply if the product has been altered or repaired by anyone other than a Mattel Electronics authorized service center or if the product has been subjected to purchaser abuse, accident, negligence, or damage subsequent to purchase. This warranty gives you specific rights and you may also have other rights which vary from state to state.

You may write us, but **DO NOT SEND PRODUCT FOR REPAIR,** at the following address:

Mattel Electronics, 5000 West 147th Street,
Hawthorne, California 90250

Units returned without proof of the date of purchase or units returned after the 90 day warranty period has expired, will be repaired or replaced (at our option) for a service charge. Call one of the following telephone numbers to obtain the location of the nearest service center and the amount of the repair charge.

In the **EASTERN UNITED STATES (800) 257-5185;**

NEW JERSEY RESIDENTS MUST CALL (609) 655-3533
direct or collect;

In the **WESTERN UNITED STATES (800) 421-2826;**

ALASKA, CALIFORNIA, or **HAWAII RESIDENTS MUST CALL (213) 978-6850** direct or collect.

ATTENTION MILITARY PERSONNEL, if your Intellivision® COMPUTER KEYBOARD requires service while on duty in a foreign country, you should contact the military exchange serving your area.

IF YOU NEED SERVICE, CALL ONE OF THE SERVICE/INFORMATION NUMBERS ABOVE.

This warranty does not apply if the product has been altered or repaired by anyone other than a Mattel Electronics authorized service center or if the product has been subjected to purchaser abuse, accident, negligence, or damage subsequent to purchase. This warranty gives you specific rights and you may also have other rights which vary from state to state.

You may write us, but **DO NOT SEND PRODUCT FOR REPAIR,** at the following address:

Mattel Electronics, 5000 West 147th Street,
Hawthorne, California 90250

SPECIAL NOTICE

WARNING: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

This equipment generates and uses radio frecuency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

Reorient the receiving antenna.

Relocate the computer with respect to the receiver.

Move the computer away from the receiver.

Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

"How to Identify and Resolve Radio-TV Interference Problems".

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4.

IMPORTANT

Before you meet your new computer, please note this necessary disclaimer:

Mattel Electronics does not assume any liability or responsibility for loss or damage, direct or indirect, caused by any software programs (whether sold by Mattel Electronics or otherwise) or the use made of any such programs by the consumer.

NOTICE — PROJECTION TV OWNERS

Some stationary game patterns produced by this product may be permanently imprinted on Projection TV tubes by extended use at high brightness levels. Consult Projection TV Owner's manual before use of this product.

Model No.______________________________

Serial No.______________________________

INTELLIVISION® MATTEL ELECTRONICS®

ATTENTION !

The following pages contain important information for you, regarding:

- Errors in your Computer Module Owner's Manual.
- Cartridge requirements for use of your Computer Module.
- Cassette recorder recommendations.

Errata

The following errors escaped our proofreaders and are hiding out in the pages of your Owner's Manual. The notes below will help you ferret out these pesky critters and "debug" your manual.

To guide you to their location, each error is referenced by page number and column.

Page	Column	Error
11	Left-Top	Step 4 instructs you to stop the program by pressing the ESC key. To stop the SOUND, type CALL HUSH and press RTN
20	Left-Bottom	The arrow pointing to the Space Bar refers to copy which states that a blank space typed with the space bar uses memory. This is not true.
21	Left-Center	Again, the book states that a blank space typed with the space bar uses memory. Not so!
41	Left-Bottom	The second program line reads SP(O) = 20. It should read SQ(O) = 20.
49	Right-Center	The book states that the maximum number of elements in an array is 250, numbered from 1 to 250. The correct maximum number is 251, numbered from 0 to 250.
54	Left-Bottom	The last line reads: 10 INPUT "ENTER NUMBER", A. It should read: 10 INPU "ENTER NUMBER", A.
56	Left-Bottom & Right-Top	The command CSAV is miss-pelled CSAVE twice on this page.
68	Left-Center	The first program line reads: YP(O) = 10. It should read: XP(O) = 10.

Page	Column	Error
71	Right-Bottom	After typing in the sample program given, type CALL GRAB and press RTN to display a moving object. Then RUN the sample program, using the Disc on a Hand Controller to move the object.
72	Right-Top	Change the example program to read: 10 O = 0 (Letter O = zero) 20 CALL GRAB (Displays object zero) 30 O = 1 (Letter O = one) 40 CALL GRAB (Displays object 1) 50 CALL LINK (Links objects zero and 1) 60 XV(1) = 10 70 YV(1) = 5 80 XV(0) = 10
	Right-Bottom	We forgot to mention the procedure for UNlinking two objects. Set O = -1. Then type CALL LINK again.
73	Left-Top	Line 10 in the example program should read: 10 C = 2
73	Right-Top	Add a line to the beginning of the sample program. This line should read: 5 V = 12 (This sets the volume of the tone to be generated.)
75	Right-Top	The example: A$ = "Hello" should read: SET A$ = "HELLO".
77	Left-Center	Again, the maximum number of elements in an array is 251, numbered from 0 to 250.
84	Left-Top	In the second paragraph, we state that "most of the computer's internal operating instructions are stored in ROM, as well as the built-in BASIC and the "interpreter" that translates BASIC into the assembly language that the computer understands." A more accurate statement would be that the computer's internal operating instructions, including the build-in BASIC interpreter, are stored in ROM.

Page	Column	Error
85	Left-Center	Attempts to divide by zero are not really refused, but rather NOTED with a short whistle.
	Left-Bottom	The last paragraph is incorrect. Ignore it.
	Right-Center	The example reads: SET A$ = "Hello". It should read: SET A$ = "HELLO".
87	Right-Center	The definition of ASSEMBLY LANGUAGE is not accurate nor relevant to this system. Consider instead the following definition of MACHINE LANGUAGE: A binary language (1's and 0's) which is the only language a digital computer recognizes.

Special Notes

USE A CARTRIDGE WITH THE COMPUTER MODULE. In order to use the Computer Module, a cartridge must be plugged into the Adaptor's cartridge port. If you only wish to use the built-in BASIC for programming, plug in any standard cartridge. A standard cartridge has a title screen that looks like this:

Mattel Electronics
presents
TITLE
Copr 19xx Mattel

If a cartridge isn't plugged in, nothing will appear on the TV screen.

CASSETTE RECORDER SPECIFICATIONS. There are many acceptable recorders in the marketplace today. Your Computer Adaptor requires the following electrical specifications from a data or video-recorder:

1) Minimum frequency response of 500 Hz to 5000 Hz +/- 3db.
2) Minimum acceptable output level of 3 volts peak to peak at 2400 Hz and 4800 Hz into a 100 OHM load.
3) 1-7/8 inch per second speed, +/- 3%.

Recommended Recorders:

Aquarius Data Recorder
Toshiba KT-P22
Toshiba KT 1500
Realistic CTR-57
Realistic CTR-60
Realistic Minisette 9-Model 14-812
Sears 799-217-21801
Radio Shack TRS 80-26-1208
Radio Shack TRS 80 CTR 56 (Note: This recorder has no counter.)
GE 3-5009

No. 4187-0170 PRINTED IN HONG KONG